

Oracle® Beehive

Installation Guide

Release 1 (1.2 and 1.3) for Solaris Operating System (SPARC
64-Bit)

E10482-03

August 2008

Oracle Beehive Installation Guide, Release 1 (1.2 and 1.3) for Solaris Operating System (SPARC 64-Bit)

E10482-03

Copyright © 2008, Oracle. All rights reserved.

Primary Author: Raymond Gallardo

Contributing Author: Jay Rajiva

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	xiii
Audience	xiii
Documentation Accessibility	xiii
Related Documents	xiv
Conventions	xiv
1 What You Should Know Before Installing Oracle Beehive	
Requirements	1-1
Operating System Certification	1-1
Required Operating Systems Packages for SPARC	1-1
Required Operating System Patches	1-2
Set Shell Limits for the oracle User	1-3
Database Requirements	1-4
Oracle Enterprise Manager Grid Control Requirements	1-4
Hardware Requirements	1-4
Installation Scenarios	1-4
Installing Oracle Beehive Against an Existing Database	1-5
Installing Oracle Beehive Provisioning Application to Allow Oracle Beehive to Be Provisioned through Oracle Enterprise Manager Grid Control 1-5	
Specifying Location of Oracle Beehive Shiphome Binaries in Grid Control	1-5
Installing Oracle Beehive for Demilitarized Zone (DMZ)	1-6
Installing Oracle Beehive Integration for Zimbra	1-7
Requirements	1-8
Installing Oracle Beehive Administration Console	1-8
Requirements and Certifications	1-8
Installing Oracle Beehive in Silent Mode	1-8
Installing Oracle Beehive in High Availability Environment	1-9
Installing More Than One Instance	1-9
Upgrading Oracle Beehive	1-9
Starting Oracle Beehive Install Wizard	1-9
Post-Installation Procedures	1-10
Cloning Application Tiers and Deployments	1-10
Uninstalling Oracle Beehive	1-11

2 Oracle Beehive Database Requirements

General Requirements	2-1
Patches	2-2
Initialization Parameters.....	2-3
Logging Settings.....	2-4
Creating a Database	2-4
Using Oracle Real Application Clusters Database	2-4
Customizing Oracle Beehive Tablespaces Layout	2-4
Using Raw Storage.....	2-5

3 Oracle Beehive Install Wizard Checklist

Oracle Beehive (Standard Installation).....	3-1
Oracle Beehive Provisioning Application.....	3-2
Oracle Beehive for DMZ.....	3-2
Oracle Beehive Integration for Zimbra.....	3-2
Oracle Beehive Administration Console	3-2

4 Oracle Beehive Install Wizard

Product Selection Sequence of Screens	4-1
Welcome	4-1
New Inventory Location	4-2
Select Product Type	4-2
Oracle Beehive Release 1.....	4-3
Provisioning Application for Oracle Beehive Release 1.....	4-3
Oracle Beehive Release 1 for DMZ.....	4-3
Oracle Beehive Integration for Zimbra	4-3
Oracle Beehive (Standard Installation) Sequence of Screens	4-4
Specify Home Location	4-4
Specifying Oracle Home Directory	4-4
Prerequisite Checks.....	4-5
Select Installation Type.....	4-6
Database Information	4-7
Enterprise Definition Information.....	4-8
Existing Enterprise Information.....	4-9
Schema Password.....	4-9
Existing Schema Password	4-10
beeadmin Password.....	4-10
Installation Summary	4-11
Progress Indicator	4-12
Configuration Assistants.....	4-12
End of Installation	4-12
Oracle Beehive Provisioning Application Sequence of Screens.....	4-12
Specify Home Location	4-13
Repository Database Administrator Password	4-13
Installation Summary	4-13
Progress Indicator	4-14

Configuration Assistants.....	4-14
End of Installation.....	4-14
Oracle Beehive for DMZ Sequence of Screens.....	4-14
Specify Home Location	4-15
Prerequisite Checks.....	4-15
Installation Summary	4-15
Progress Indicator	4-15
Configuration Assistants.....	4-16
End of Installation.....	4-16
Oracle Beehive Integration for Zimbra Sequence of Screens.....	4-16
Specify Home Location	4-17
Prerequisite Checks.....	4-17
Select Installation Type.....	4-17
Database Information	4-18
Existing Enterprise Information.....	4-20
Existing Schema Password	4-20
Installation Summary	4-20
Progress Indicator	4-21
Configuration Assistants.....	4-21
End of Installation.....	4-22

5 Oracle Beehive Administration Console Installation Help

Oracle Beehive Administration Console Sequence of Screens.....	5-1
Welcome	5-1
New Inventory Location	5-2
Specify Home Location	5-2
Prerequisite Checks.....	5-3
Database Information	5-3
Installation Summary	5-4
Installation Process	5-5
Configuration Assistants.....	5-5
End of Installation.....	5-5
Starting the Oracle Beehive Administration Console Uninstall Wizard	5-5
Oracle Beehive Administration Console Uninstall Wizard Sequence of Screens	5-5
Select Oracle Home to Uninstall	5-5
Start Uninstallation	5-6
End of Uninstallation.....	5-6

6 Upgrading Oracle Beehive

Oracle Beehive Upgrade Process Sequence of Screens.....	6-1
Upgrade Existing Home.....	6-2
Specify Home Location to Upgrade	6-2
Shutdown Instances.....	6-2
Database Information for Upgrade	6-2
Upgrade Summary.....	6-2
Upgrade Process.....	6-3

Configuration Assistants.....	6-3
End of Upgrade	6-3
Oracle Beehive Provisioning Application Upgrade Process Sequence of Screens	6-3
Upgrade Existing Home.....	6-4
Specify Home Location to Upgrade	6-4
Database Information for Upgrade	6-4
Upgrade Summary.....	6-4
Upgrade Process.....	6-5
Configuration Assistants.....	6-5
End of Upgrade	6-5
Oracle Beehive for DMZ Upgrade Process Sequence of Screens	6-5
Upgrade Existing Home.....	6-5
Specify Home Location to Upgrade	6-5
Upgrade Summary.....	6-6
Upgrade Process.....	6-6
Configuration Assistants.....	6-6
End of Upgrade	6-6

7 Installing Oracle Beehive with Provisioning Application

Step 1: Zip Installable Version of Oracle Beehive	7-2
Step 2: Upload Zipped Installable Version of Oracle Beehive to Oracle Grid Control Software Library	7-2
Step 3: Run Oracle Beehive Provisioning Application.....	7-3
Uploading Oracle Beehive Shiphome to Oracle Grid Control.....	7-3
Uninstalling Oracle Beehive Provisioning Application.....	7-3

8 Running Oracle Beehive Provisioning Application

Introduction to Oracle Beehive Provisioning Application	8-1
Running Oracle Beehive Provisioning Application.....	8-2
Target List Page.....	8-3
Install Inputs Page.....	8-3
Credentials/Schedule Page	8-5
Summary Page.....	8-6

9 Installing Oracle Beehive in Silent Mode (Non-Interactive)

Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard	9-1
Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard and Config Wizard.....	9-3
Installing Oracle Beehive Provisioning Application in Silent Mode.....	9-4
Installing Oracle Beehive for DMZ in Silent Mode.....	9-4
Verifying Oracle Beehive Installation.....	9-5
Recovering from Failed Oracle Beehive Configuration	9-5
Upgrading Oracle Beehive in Silent Mode	9-5
Uninstalling Oracle Beehive in Silent Mode.....	9-6

10	Oracle Beehive Install and Config Wizard Command-Line Options	
	Oracle Beehive Install Wizard	10-1
	Usage.....	10-1
	Options	10-1
	Oracle Beehive Config Wizard	10-2
	Usage.....	10-2
	Options	10-3
11	Oracle Beehive Uninstall Wizard	
	Starting the Oracle Beehive Install Wizard.....	11-1
	Oracle Beehive Uninstall Wizard Sequence of Screens.....	11-1
	Select Oracle Home to Uninstall	11-1
	Option to Remove Database Schema	11-2
	Start Uninstallation	11-2
	End of Uninstallation.....	11-2
	Uninstalling Multiple Instances	11-2
	Manually Uninstalling Oracle Beehive	11-3
	Manually Deleting Oracle Beehive Tablespaces and Datafiles	11-3
	Manually Deleting Customized Oracle Beehive Customized Tablespaces and Datafiles ...	11-5
12	Oracle Beehive Post-Installation Procedures	
	Using Oracle Beehive Command-Line Utility.....	12-1
	Configuring Oracle Beehive to Listen on Ports Less Than 1024	12-1
	Changing HTTP Port	12-2
	Changing Other Ports.....	12-3
	Enabling BPEL Workflows to Use New Port Number	12-4
	Opening Ports Required by Oracle Beehive for FTP.....	12-5
	Configuring DMZ Instances	12-7
	Integrating and Synchronizing LDAP with Oracle Beehive	12-7
	Configuring SSL.....	12-7
	Configuring TLS.....	12-7
	Configuring SSL for LDAP Integration.....	12-7
	Enabling AJPS.....	12-8
	Configuring Oracle Beehive E-mail.....	12-8
	Configuring Oracle Secure Enterprise Search	12-8
	Configuring Oracle Single Sign-On.....	12-8
	Configuring External Oracle BPEL Process Manager with Oracle Beehive	12-8
	Installing Oracle Beehive Integration for Outlook	12-8
	Configuring Oracle Beehive Integration for Zimbra	12-9
	Cloning Oracle Beehive	12-10
	Performing Oracle Beehive Administration Console Post-Installation Procedures.....	12-10
13	Configuring Oracle Beehive Demilitarized Zone Instances	
	Step A: Configuring Oracle Wallet with Oracle Beehive DMZ Instances	13-1
	Step B: Configuring Oracle Beehive DMZ Instances.....	13-2

Troubleshooting DMZ Configuration.....	13-5
14 Integrating and Synchronizing LDAP with Oracle Beehive	
How Synchronization Works	14-2
Requirements	14-2
Synchronizing LDAP with Oracle Beehive User Directory Service	14-2
Step 1: Creating an LDAP Mapping Profile	14-3
Step A: Creating an LDAP Mapping Profile from a Template.....	14-3
Step B: Renaming the Profile	14-4
Step C: Specifying LDAP Server Settings.....	14-4
Step D: Providing Mapping Details for Each User Type and Static Group	14-5
Exclusion and Inclusion	14-6
Step E: Providing Scope and Membership Mapping Information	14-6
Step F: Providing Attribute Mapping for Each User Type and Static Group	14-8
Mapping Active Directory Proxy Addresses.....	14-8
Step G: Adding Profile to Oracle Beehive	14-9
Directory Profile Validation	14-9
Step 2: Enabling Synchronization	14-10
Step 3: Loading Users and Groups	14-11
Controlling How Often UDS Contacts the LDAP Server.....	14-12
Retrieving Information About the LDAP Server.....	14-13
Default UserObjectClass and GroupObjectClass Values	14-15
Configuring Authentication Service to Use LDAP Server	14-15
Configuring Digest Authentication	14-16
Step A: Determine Digest Mechanism Depending on LDAP Directory	14-16
Step B: Configure Oracle Beehive	14-17
Changing LDAP Administrator's Password	14-18
Oracle Internet Directory Considerations	14-19
Synchronizing with Directory Replication Group	14-19
Migrating Oracle Internet Directory from One Server to Another	14-19
Troubleshooting Synchronization between Oracle Beehive and Oracle Internet Directory.....	14-20
Active Directory Considerations	14-20
Troubleshooting General LDAP Synchronization Issues	14-21
15 Configuring SSL	
SSL Checklist	15-1
Configuring SSL with Oracle Beehive	15-1
Configuring SSL with Test Certificates for Oracle Beehive	15-2
Configuring SSL with Real Certificates During Installation of Oracle Beehive.....	15-3
Configuring SSL with Real Certificates After Installation of Oracle Beehive	15-4
Configuring SSL with Oracle Beehive DMZ Instances	15-5
Configuring SSL with Test Certificates After Installation of DMZ Instances	15-5
Configuring SSL with Real Certificates After Installation of DMZ Instances.....	15-5
Procedures Related to Configuring SSL	15-6
Configuring Oracle Wallet for Oracle Database and Oracle RAC.....	15-6
Creating Self-Signed Certificate and Importing it into Wallet	15-6

Using Oracle Wallet to Create Self-Signed Certificate	15-7
Creating CA-Signed Certificate and Importing it into Wallet.....	15-7
16 Configuring TLS with Oracle Wallet	
Step 1: Creating Auto-Login Oracle Wallet	16-1
Step 2: Configuring Oracle Beehive Instance to Use Oracle Wallet.....	16-1
Changing Oracle Wallet Password.....	16-2
Configuring TLS on Multiple Instances.....	16-2
Enabling ORMIS with Password-Protected Oracle Wallet.....	16-2
Disable ORMIS	16-2
Enable ORMIS with Default SSL.....	16-3
Enable ORMIS with Password Protection.....	16-3
17 Configuring SSL for LDAP Integration	
Requirements	17-1
Configure SSL for LDAP Verification.....	17-2
Configure Apache HTTP Server for WebDAV Folders	17-3
18 Enabling AJPS	
Enabling AJPS.....	18-1
Disabling AJPS	18-3
19 Configuring E-Mail with SSL	
Securing E-Mail with SSL	19-1
Enabling Both Non-SSL and SSL Ports	19-2
Modifying an Endpoint in SMTP	19-2
Adding an Endpoint to SMTP	19-3
Modifying or Adding an Endpoint in IMAP.....	19-5
Creating and Configuring an Endpoint.....	19-5
Configuring E-Mail Client	19-6
20 Configuring Oracle Secure Enterprise Search	
Step A: Configure Oracle Beehive	20-1
Step B: Configure Oracle Secure Enterprise Search.....	20-2
21 Configuring Single Sign-On with Oracle Beehive	
22 Configuring External Oracle BPEL Process Manager with Oracle Beehive	
Step A: Configure Oracle Beehive	22-1
Step B: Configure External Oracle BPEL Process Manager	22-2
Deploying Custom Workflows with External Oracle BPEL Process Manager.....	22-3
23 Installing Oracle Beehive Integration for Outlook	
Requirements	23-1

Installing Oracle Beehive Integration for Outlook Using the Remote Downloader	23-1
Pre-Seeding a Server Name in the Remote Downloader	23-2
Pre-Seeding Port Number and Secure Connection.....	23-2
Installing Oracle Beehive Integration for Outlook Interactively	23-2
Installing Oracle Beehive Integration for Outlook Silently	23-2
Uninstalling Oracle Beehive Integration for Outlook	23-3
Installing Oracle Beehive Integration for Outlook Using Active Directory	23-3
Creating an msi installation package for Oracle Beehive Integration for Outlook	23-3
Deploying Oracle Beehive Integration for Outlook using Active Directory Group Policy .	23-3
Verifying that Oracle Beehive Integration for Outlook is available for installation	23-4
Updating Oracle Beehive Integration for Outlook in the DMS Repository	23-4
Applying the Language Pack for Oracle Beehive Integration for Outlook	23-5
Applying Configuration Settings from the DMS	23-7
Customizing General Installation Settings	23-7
Downgrading your Oracle Beehive Integration for Outlook Installation	23-8
Registry Settings	23-8
INI File Settings	23-9
Enforcing Upgrades	23-9
Allowing a Grace Period Before Enforcing Upgrades.....	23-9
Sample XML Provisioning File	23-10

24 Cloning Oracle Beehive Application Tiers and Sites

Introduction to Cloning	24-1
Preparing Source Application Tier Instance	24-4
Step 1: Verify Requirements	24-4
Step 2: Clear or Activate Any Pending Configuration Changes to the Central Configuration Repository	24-4
Step 3: Unset Environment Variables.....	24-5
Step 4: Shut Down All Processes On the Application Tier	24-5
Step 5: Call beectl clone_preparation Command	24-5
Step 6: Zip Files to Create Clone Image	24-6
Application Tier Cloning	24-6
Step 1: Unzip Compressed Oracle Home	24-7
Step 2: Set PERL5LIB Environment Variable	24-7
Step 3: Modify Oracle Home Path	24-8
Step 4: Execute beectl clone_midtier Command.....	24-8
Step 5: Perform Miscellaneous Operations	24-10
Site Cloning	24-10
Step 1: Unzip Compressed Oracle Home	24-10
Step 2: Set PERL5LIB Environment Variable	24-11
Step 3: Modify Oracle Home Path	24-11
Step 4: Execute beectl clone_site Command.....	24-11
Step 5: Prevent Services from Target Application Tiers from Referring to External Resources	24-15
Stopping User Directory Service from Referring to LDAP Server	24-15
Stopping Authentication Service from Referring to LDAP Server.....	24-16
Stopping Virus Scanner Process from Referring to External Virus Scan Engine	24-16

Step 6: Perform Miscellaneous Operations	24-16
OC4J Instance Cloning.....	24-16
Oracle Beehive Integration for Zimbra Cloning.....	24-17
Customizing Files or Directories in a Cloned Image	24-17
Customizing Ports in a Cloned Instance.....	24-18
Cloning Application Tiers and Sites with Ports Less Than 1024.....	24-18
Oracle Inventory Location Option of Clone Commands on UNIX-Based Systems.....	24-18
Cloned Application Tiers Are Not Automatically SSL or AJPS Enabled	24-19
Cloned Application Tiers and LDAP Synchronization	24-19
Replicating LDAP Server for Cloned Instance.....	24-19
Testing Replicated LDAP in Cloned Instance.....	24-22
Troubleshooting Replicated LDAP.....	24-22
Site Cloning and Multiple Instances.....	24-22
References to Oracle Application Server Cloning Documentation	24-23
25 Oracle Beehive Administration Console Post-Installation Procedures	
Configuring Oracle Beehive Administration Console for SSL Access.....	25-1
Changing Oracle Beehive Administration Console Port.....	25-2
Extending Java Single Sign-On Session Timeout.....	25-2
Configuring Oracle Beehive Administration Console for LDAP-Based Authentication	25-3
Configuring Virtual Host.....	25-4
Restarting OC4J	25-4
26 Configuring and Installing Oracle Beehive for Oracle RAC	
Naming Conventions.....	26-1
Pre-Install Steps.....	26-2
Installation.....	26-3
Post-Install Steps	26-3
Adding Oracle RAC Nodes to an Existing Oracle Beehive Instance	26-6
Installing Multiple Oracle Beehive Instances with Oracle RAC.....	26-7
Install All Oracle Beehive Instances, then Apply Post-Install Steps.....	26-7
Install One Oracle Beehive Instance, Apply Post-Install Steps, then Add Additional Instances...	26-7
27 Installing Oracle Beehive in High Availability Environment	
Configuring High Availability Environment with Load Balancer.....	27-1
Configuring High Availability Environment with a DMZ Instance	27-3
Configuring High Availability Environment with DMZ Instances and Load Balancer.....	27-4

Preface

The *Oracle Beehive Installation Guide for Linux* describes how to install and configure Oracle Beehive.

Audience

The *Oracle Beehive Installation Guide for Linux* is directed at any administrator whose task is the installation, deployment, configuration, administration, and maintenance of Oracle Beehive.

Documentation Accessibility

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

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documents

For more information, see the following documents in the Oracle Beehive Release 1 documentation library:

- *Oracle Beehive Administrator's Guide*
- *Oracle Beehive Administrator's Reference Guide*
- *Oracle Beehive Application Developer's Guide*
- *Oracle Beehive Concepts*
- *Oracle Beehive Deployment Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	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

Oracle Beehive Installation

This part describes how to install and uninstall Oracle Beehive. To configure Oracle Beehive, refer to [Part II, "Oracle Beehive Post-Installation Configuration"](#). To install Oracle Beehive in a high availability environment, refer to [Part III, "Oracle Beehive High Availability Configuration"](#).

This part contains the following modules:

- [What You Should Know Before Installing Oracle Beehive](#)
- [Oracle Beehive Database Requirements](#)
- [Oracle Beehive Install Wizard Checklist](#)
- [Oracle Beehive Install Wizard](#)
- [Oracle Beehive Administration Console Installation Help](#)
- [Upgrading Oracle Beehive](#)
- [Installing Oracle Beehive with Provisioning Application](#)
- [Running Oracle Beehive Provisioning Application](#)
- [Installing Oracle Beehive in Silent Mode \(Non-Interactive\)](#)
- [Oracle Beehive Install and Config Wizard Command-Line Options](#)
- [Oracle Beehive Uninstall Wizard](#)

What You Should Know Before Installing Oracle Beehive

This module describes the hardware and software prerequisites of Oracle Beehive and available installation scenarios. It covers the following topics:

- [Requirements](#)
- [Installation Scenarios](#)
- [Upgrading Oracle Beehive](#)
- [Starting Oracle Beehive Install Wizard](#)
- [Post-Installation Procedures](#)
- [Cloning Application Tiers and Deployments](#)
- [Uninstalling Oracle Beehive](#)

Requirements

This section describes the following Oracle Beehive software and hardware requirements:

- [Operating System Certification](#)
- [Required Operating Systems Packages for SPARC](#)
- [Required Operating System Patches](#)
- [Set Shell Limits for the oracle User](#)
- [Database Requirements](#)
- [Oracle Enterprise Manager Grid Control Requirements](#)
- [Hardware Requirements](#)

Operating System Certification

Oracle Beehive is certified for Sun Solaris 10 (SPARC 64-bit).

Required Operating Systems Packages for SPARC

Oracle Beehive requires the following packages on Solaris operating systems (SPARC 64-Bit):

- SUNWbtool
- SUNWhea

- SUNWlibm
- SUNWlibms
- SUNWspot
- SUNWtoo
- SUNWi1of
- SUNWxwfont

Required Operating System Patches

Table 1-1 lists the Solaris operating system patches that you must install before installing Oracle Beehive. These are patches that make up the "J2SE patch cluster". You can download the patches from:

<http://sunsolve.sun.com>

Refer to the README file included with the patch for instructions on installing it.

Note: Although the patches listed in Table 1-1 are current at the time of publication, Sun Microsystems, Inc. often updates the list of required patches for J2SE. Click the "Readme" link on the following URL for the most current list of patches:

<http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage>

Table 1–1 Patches Required for Solaris Operating System (SPARC 64-Bit)

Operating System	Patches Required
Solaris 8	<ul style="list-style-type: none"> ■ 108652-82 or higher: X11 6.4.1: Xsun patch ■ 108921-21 or higher: CDE 1.4: dtwm patch ■ 108940-62 or higher: Motif 1.2.7 and 2.1.1: Runtime library patch ■ 108773-18 or higher: IIM and X input and output method patch ■ 111310-01 or higher: /usr/lib/libdhcpagent.so.1 patch ■ 109147-28 or higher: Linker patch ■ 111308-04 or higher: /usr/lib/libmtmalloc.so.1 patch ■ 112438-03 or higher: /kernel/drv/random patch ■ 108434-17 or higher: 32-bit shared library patch for C++ ■ 108435-17 or higher: 64-bit shared library patch for C++ ■ 111111-04 or higher: /usr/bin/nawk patch ■ 112396-02 or higher: /usr/bin/fgrep patch ■ 110386-03 or higher: RBAC feature patch ■ 111023-03 or higher: /kernel/fs/mntfs and /kernel/fs/sparcv9/mntfs patch ■ 111317-05 or higher: /sbin/init and /usr/sbin/init patch ■ 113648-03 or higher: /usr/sbin/mount patch ■ 115827-01 or higher: /sbin/sulogin and /sbin/netstrategy patch ■ 116602-01 or higher: /sbin/uadmin and /sbin/hostconfig patch ■ 108987-13 or higher: Patch for patchadd and patchrm ■ 108528-29 or higher: Kernel update patch ■ 108989-02 or higher: /usr/kernel/sys/acctctl and /usr/kernel/sys/exacctsys patch ■ 108993-36 or higher: LDAP2 client, libc, libthread and libnsl libraries patch ■ 109326-14 or higher: libresolv.so.2 and in.named patch ■ 110615-11 or higher: sendmail patch
Solaris 9	<ul style="list-style-type: none"> ■ 113096-03 or higher: X11 6.6.1: OWconfig patch ■ 112785-35 or higher: X11 6.6.1: Xsun patch
Solaris 10	None

Set Shell Limits for the oracle User

Kernel parameter values are stored in the `/etc/system` file. If your `/etc/system` file does not specify the parameters, you have to add lines to the file to specify them (on Solaris 8 only).

Parameter	Line in /etc/system Showing the Minimum Value	Description
rlim_fd_max	set rlim_fd_max=65536	Defines the hard limit on file descriptors that a single process can have open. To determine the current setting, run this command: prompt> ulimit -Hn

Database Requirements

Refer to "[Oracle Beehive Database Requirements](#)" for information about Oracle Beehive database requirements, creating databases, and configuring database settings.

Note: The installation scenario "[Installing Oracle Beehive for Demilitarized Zone \(DMZ\)](#)" does not require a database. See this section for more information about this scenario.

Oracle Enterprise Manager Grid Control Requirements

If you want to install the Oracle Beehive Provisioning Application, which allows you to install Oracle Beehive from Oracle Grid Control, you must have Oracle Enterprise Manager 10g Release 4 Grid Control (10.2.0.4) or later.

Hardware Requirements

The following are the minimum hardware requirements for Oracle Beehive:

- Disk Space: 2.6 GB
- Swap Space: 2,048 MB
- Memory:
 - Oracle Beehive: 2 GB RAM
 - Oracle Beehive for DMZ: 1 GB

Notes: The values for minimum disk space and memory do not include the amounts required for the target database.

Oracle Beehive does not support IPv6 (Internet Protocol version 6). You must disable IPv6 from the server on which you plan to install Oracle Beehive before starting the installation process.

Installation Scenarios

This section describes the possible ways you may install Oracle Beehive. It covers the following topics:

- [Installing Oracle Beehive Against an Existing Database](#)
- [Installing Oracle Beehive Provisioning Application to Allow Oracle Beehive to Be Provisioned through Oracle Enterprise Manager Grid Control](#)
- [Installing Oracle Beehive for Demilitarized Zone \(DMZ\)](#)

- [Installing Oracle Beehive Integration for Zimbra](#)
- [Installing Oracle Beehive Administration Console](#)
- [Installing Oracle Beehive in Silent Mode](#)
- [Installing Oracle Beehive in High Availability Environment](#)
- [Installing More Than One Instance](#)

Installing Oracle Beehive Against an Existing Database

This is the standard installation scenario that will give you a fully functioning instance of Oracle Beehive.

This scenario gives you the choice to install and configure Oracle Beehive, or to install Oracle Beehive now and configure it later.

For more information about installing Oracle Beehive against an existing database, refer to "[Oracle Beehive \(Standard Installation\) Sequence of Screens](#)".

Installing Oracle Beehive Provisioning Application to Allow Oracle Beehive to Be Provisioned through Oracle Enterprise Manager Grid Control

Oracle Enterprise Manager Grid Control (Oracle Grid Control) offers a centralized environment with which you can manage the complete Oracle IT infrastructure, including systems running Oracle and non-Oracle technologies. Oracle Grid Control gives you a broad set of administration, configuration management, provisioning, end-to-end monitoring, and security capabilities.

Oracle Beehive Provisioning Application allows you to install and configure Oracle Beehive instances in the robust environment of Oracle Grid Control.

Oracle Beehive Provisioning Application allows you to upload an installable version of Oracle Beehive into the central software library of Oracle Grid Control. Oracle Beehive Provisioning Application also creates a deployment procedure with which you can install and configure Oracle Beehive on any host through the deployment procedure manager of Oracle Grid Control.

For more information about installing Oracle Beehive Provisioning Application, refer to "[Oracle Beehive Provisioning Application Sequence of Screens](#)". For more information about the Oracle Beehive Provisioning Application, refer to "[Installing Oracle Beehive with Provisioning Application](#)".

Specifying Location of Oracle Beehive Shiphome Binaries in Grid Control

Before you install Oracle Beehive Provisioning Application, you must specify the location where Oracle Grid Control will look for the Oracle Beehive shiphome binaries, which is a zip file containing an installable version of Oracle Beehive. The following steps show you how to do this:

1. Navigate to Oracle Grid Control, `http://<Oracle Grid Control host>:<Oracle Grid Control port>/em`.
2. Click the **Deployments** tab. Click **Provisioning** from the menu bar. Click **Administration**. In the section **Software Library Configuration**, specify the location for the software library. This is a directory where Oracle Grid Control will look for the Oracle Beehive shiphome binaries.

Note: If you have more than one Oracle Management Service sharing the same management repository, make sure each Oracle Management Service can read from and write to the location of the software library.

Installing Oracle Beehive for Demilitarized Zone (DMZ)

This scenario installs and configures Oracle Beehive into a demilitarized zone (DMZ) that will integrate with the rest of the Oracle Beehive system.

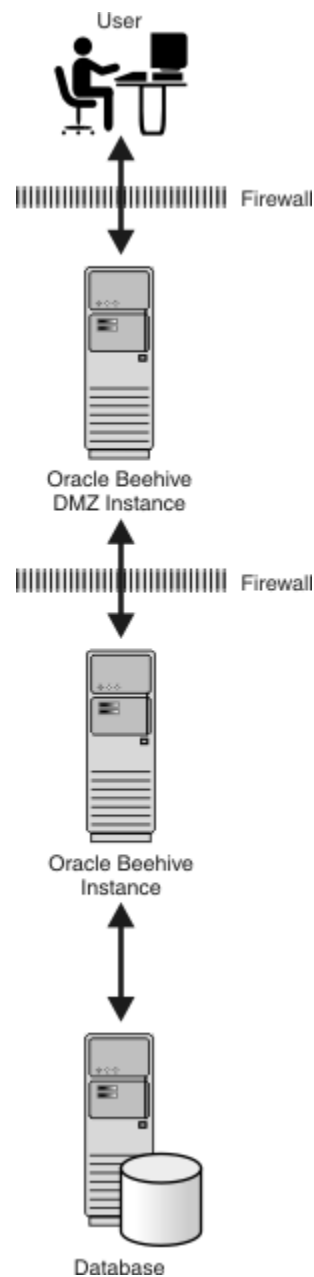
An Oracle Beehive DMZ configuration consists of one or more Oracle Beehive DMZ instances that are isolated from the Internet by a firewall on one side, and from one or more Oracle Beehive non-DMZ instances by a firewall on the other side. The DMZ instances are viewed as semi-secure. They are protected from the open Internet, but are not completely trusted like the non-DMZ instances that are inside the second firewall and part of the company's intranet. Oracle Beehive data is stored only in the non-DMZ instances, and administrative tasks may only be performed on the non-DMZ instances.

Oracle Beehive for DMZ requires an existing (non-DMZ) Oracle Beehive instance.

Oracle Beehive for DMZ does not require a database. However, the administration tool `beectl` will be disabled. Because this installation scenario does not require a database, you only have to specify a target installation location, which the Oracle Beehive Install Wizard will configure as a DMZ.

The following image illustrates an Oracle Beehive DMZ instance integrated in an Oracle Beehive system:

Figure 1-1 Oracle Beehive DMZ Instance Integrated in Oracle Beehive System



For more information about installing Oracle Beehive for a DMZ, refer to "[Oracle Beehive for DMZ Sequence of Screens](#)".

Installing Oracle Beehive Integration for Zimbra

Oracle Beehive Integration for Zimbra is available for Oracle Beehive Release 1 (1.3) and later.

Oracle Beehive Integration for Zimbra is a Web-based client for Oracle Beehive. For more information about installing this product, refer to "[Oracle Beehive Integration for Zimbra Sequence of Screens](#)".

After installing this product, refer to "[Configuring Oracle Beehive Integration for Zimbra](#)".

Requirements

The following are the minimum hardware requirements for Oracle Beehive Integration for Zimbra:

- Disk Space: 1.3 GB
- Memory: 1 GB RAM

The following Web browsers are supported for Oracle Beehive Integration for Zimbra:

- Microsoft Windows XP and Vista: Internet Explorer 7 and 6.0 SP2 and Firefox 1.5 and later and 2.0
- Fedora Core 4: Firefox 1.5 and 2.0
- Mac OS X 10.4: Firefox 1.5 and 2.0

Installing Oracle Beehive Administration Console

Oracle Beehive Administration Console is available for Oracle Beehive Release 1 (1.3) and later.

Oracle Beehive Administration Console is a secure, browser-based administration client. It provides Oracle Beehive administrators centralized and role-based access to system configuration and management, user and workspace administration, monitoring, and reporting functions. For more information about installing this product, refer to "[Oracle Beehive Administration Console Installation Help](#)".

Note: Oracle Beehive Administration Console version 1.3.1.0.0 is only supported by Oracle Beehive version 1.3.1.0.0.

Requirements and Certifications

This version of Oracle Beehive Administration Console is certified for Oracle Beehive Release 1 (1.3.1.0.0) and Web browsers Mozilla Firefox version 2 or later and Microsoft Internet Explorer version 7.

The following are the minimum hardware requirements for Oracle Beehive Administration Console:

- Disk Space: 750 MB
- Memory: 1GB RAM

Installing Oracle Beehive in Silent Mode

Silent mode allows you to install Oracle Beehive with minimal interaction. You provide the Oracle Beehive Install Wizard with a response file that contains all the required information to install and configure Oracle Beehive. As a result, you do not have to be present to select, specify, and confirm options in each step of the installation process.

This scenario is appropriate if you want to install Oracle Beehive as a batch process or do not want to use the GUI provided by the Install Wizard.

You may install Oracle Beehive against an existing database or Oracle Beehive Provisioning Application in silent mode.

For more information about installing Oracle Beehive in silent mode, refer to ["Installing Oracle Beehive in Silent Mode \(Non-Interactive\)"](#).

Installing Oracle Beehive in High Availability Environment

Installing Oracle Beehive in a high availability environment involves installing a third-party load balancer, installing multiple Oracle Beehive instances, and configuring the virtual server of each instance.

For more information, refer to ["Installing Oracle Beehive in High Availability Environment"](#).

Installing More Than One Instance

To install more than one instance of Oracle Beehive so that each instance is in its own machine and shares the same database, simply install each instance with the Install Wizard. When entering the database information for each instance, use the same global service name for the server name.

To install multiple instances of Oracle Beehive with Oracle RAC, refer to the section, ["Installing Multiple Oracle Beehive Instances with Oracle RAC"](#).

To configure a load balancer after installing multiple Oracle Beehive instances, refer to ["Installing Oracle Beehive in High Availability Environment"](#).

Upgrading Oracle Beehive

You may upgrade the following Oracle Beehive products in silent mode:

- Oracle Beehive Release 1 (1.2.1.0.0 or 1.2.2.0.0) to Release 1 (1.3.1.0.0)
- Oracle Beehive Provisioning Application version 1.2.1.0.0 to Oracle Beehive Provisioning Application version 1.3.1.0.0. (You may not upgrade any version of Oracle Beehive Provisioning Application earlier than 1.2.1.0.0.)
- Oracle Beehive for DMZ version 1.2 to version 1.3.1.0.0

Refer to ["Upgrading Oracle Beehive"](#) for information about upgrading Oracle Beehive products to version 1.3.1.0.0.

Refer to ["Upgrading Oracle Beehive in Silent Mode"](#) in ["Installing Oracle Beehive in Silent Mode \(Non-Interactive\)"](#) for more information about upgrading Oracle Beehive products in silent mode.

Starting Oracle Beehive Install Wizard

All installation scenarios use the Oracle Beehive Install Wizard.

Run `runInstaller` to start the installation and configuration of Oracle Beehive

Note: Oracle Beehive Administration Console is shipped on its own installation media (separate from Oracle Beehive). Run `runInstaller` from this installation media to start the installation and configuration of Oracle Beehive Administration Console.

The Oracle Beehive Install Wizard has several command-line options available. For more information about these options, refer to ["Oracle Beehive Install and Config Wizard Command-Line Options"](#).

Note: For all installation scenarios, except ["Installing Oracle Beehive in Silent Mode"](#), you may run the Oracle Beehive Install Wizard without any command-line options. For more information about installing Oracle Beehive in silent mode, refer to ["Installing Oracle Beehive in Silent Mode \(Non-Interactive\)"](#).

Post-Installation Procedures

Depending on the installation scenario you have chosen, your security requirements, the standards-based clients you want to use with Oracle Beehive, or any other issue particular to your deployment, perform the appropriate procedures described in ["Oracle Beehive Post-Installation Procedures"](#). These procedures include the following:

- [Using Oracle Beehive Command-Line Utility](#)
- [Configuring Oracle Beehive to Listen on Ports Less Than 1024](#)
- [Opening Ports Required by Oracle Beehive for FTP](#)
- [Configuring DMZ Instances](#)
- [Integrating and Synchronizing LDAP with Oracle Beehive](#)
- [Configuring SSL](#)
- [Configuring TLS](#)
- [Configuring SSL for LDAP Integration](#)
- [Enabling AJPS](#)
- [Configuring Oracle Beehive E-mail](#)
- [Configuring Oracle Secure Enterprise Search](#)
- [Configuring Oracle Single Sign-On](#)
- [Configuring External Oracle BPEL Process Manager with Oracle Beehive](#)
- [Installing Oracle Beehive Integration for Outlook](#)
- [Configuring Oracle Beehive Integration for Zimbra](#)
- [Cloning Oracle Beehive](#)
- [Performing Oracle Beehive Administration Console Post-Installation Procedures](#)

Cloning Application Tiers and Deployments

Cloning is the process of copying an existing installation to a different location while preserving its configuration.

You may clone an application tier, which involves preparing a "gold" image of a patched Oracle home. With this clone, you may create a new application tier with all patches applied to it in a single step. This is in contrast to separately installing, configuring, and applying any patches to Oracle Beehive.

You may also clone a deployment, which involves creating an installation that is a copy of a production, test, or development installation.

Refer to ["Cloning Oracle Beehive Application Tiers and Sites"](#) for more information.

Uninstalling Oracle Beehive

You may uninstall the following products with the Oracle Beehive Uninstall Wizard:

- Oracle Beehive Release 1
- Oracle Beehive Provisioning Application
- Oracle Beehive for DMZ
- Oracle Beehive Integration for Zimbra

Start the Oracle Beehive Uninstall Wizard by running the Oracle Beehive Install Wizard and clicking the **Uninstall** button. Alternatively, run the Oracle Beehive Config Wizard (or Oracle Beehive Install Wizard) with the `-uninstall` command-line option.

For more information about Oracle Beehive Install Wizard or Oracle Beehive Config Wizard, refer to "[Oracle Beehive Install and Config Wizard Command-Line Options](#)".

For more information about uninstalling Oracle Beehive, refer to "[Oracle Beehive Uninstall Wizard](#)".

You may uninstall Oracle Beehive in silent mode. For more information, refer to the section "[Uninstalling Oracle Beehive in Silent Mode](#)" in "[Installing Oracle Beehive in Silent Mode \(Non-Interactive\)](#)".

Oracle Beehive Database Requirements

This section describes Oracle Beehive database requirements, creating databases, and configuring database settings. It covers the following topics:

- [General Requirements](#)
- [Patches](#)
- [Initialization Parameters](#)
- [Logging Settings](#)
- [Creating a Database](#)
- [Using Oracle Real Application Clusters Database](#)
- [Customizing Oracle Beehive Tablespaces Layout](#)
- [Using Raw Storage](#)

Note: The installation scenario "[Installing Oracle Beehive for Demilitarized Zone \(DMZ\)](#)" described in "[What You Should Know Before Installing Oracle Beehive](#)" does not require a database. Refer to this section for more information about this scenario.

General Requirements

Oracle Beehive requires an existing database with the following characteristics:

- Uses Oracle Database 10g Release 2 (10.2.0.3) Enterprise Edition Patch Set 2 installed on any supported operating system.
Ensure that you have applied all the patches listed in the section "[Patches](#)".
- Uses character set AL32UTF8 (Unicode)
- Contains the following standard database components:
 - Oracle XML DB
 - Oracle *interMedia*
 - Oracle Text
 - Partitioning
- Contains Rules Manager and Expression Filter (Rules Manager and Expression Filter are installed automatically with Oracle Database 10g Enterprise Edition.)

Patches

For Microsoft Windows (32-bit and 64-bit versions), install the latest patch set for Oracle Database 10g Release 2 (10.2.0.3) Patch Set 2.

For all other operating systems, install the following patches for Oracle Database 10g Release 2 (10.2.0.3) Patch Set 2:

- Install the following patches if your database is installed on a 64-bit system:
 - 5240469 GENOCCISH GENERATES LOTS OF ERRORS AND FAILS ON AMD64. **This patch must be installed before any 64-bit patch. In addition, this patch is required only for Linux.**
 - 5252061 ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [17456], [0X69FCD9BC], [], [], [], []
- 4898608: OPATCH 10.2 ARU PLACEHOLDER
- Merge label patches:
 - 6124210: MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5924208 5705795
 - 6485648: MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5196175 AND 5472702
 - 6671491: MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 4033868 4554026 6121260
 - 6800085: MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5060402 6369463
 - 6449407: MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 6349820 6121258
 - 7031761: MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 6650096 6121248 4684070 5190392
 - 7280499: MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 6725634 6667542
- 5837795: CONNECT BY PUMP JOINING TABLE SCAN CAUSES ORA-600
- 5890966: INTERMITTENT ORA-06502 DURING PEAK LOADING
- 6150438: ORA-00932: COLLECT() DOESN'T WORK CORRECTLY FOR ADTS RESIDING IN DIFF. SCHEMAS
- 4335601: VALGRIND ERROR OF LXSCNTBY.C
- 6376915: ENQ HW - CONTENTION WITH LOB SEGMENTS
- 5524408: WRONG RESULT WHEN IN LIST HAS MORE THAN 256 BIND VARIABLES
- 7150622: MLR BUG FOR 10.2.0.3 FOR CPUJUL2008
- 6081550: TRACKING BUG FOR INVENTORY OF DG LOGICAL/LOGMINER BUGS ON TOP OF 10.2.0.3
- 6653934: XF11.1SSHR - TRC - KDRWRIC
- 5485914: MUTEX REPORTED SELF DEADLOCK AFTER DBMS_MONITOR.SESSION_TRACE_ENABLE. **This patch is required only for Solaris operating system.**

The following patches are recommended for Oracle Database 10g Release 2 (10.2.0.3) Patch Set 2:

- 5907779: "CURSOR: PIN S WAIT ON X" RUNNING DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC
- 5140631: SQL_ADDRESS AND SQL_HASH_VALUE NOT SET
- 5165885: STAR NT RAC ORA-00600 [KCLCLS_8]
- 6358116: MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5983683 6121250
- 6897966: JOBS AREN'T STARTED: SERVICE NAME IS CASE SENSITIVE

Initialization Parameters

The database requires some initialization parameters to have a minimum value, as specified in the following table:

Table 2–1 Minimum Values of Oracle Database Initialization Parameters

Property	Value
compatible	10.2.0.0.0 Note: Do not modify this parameter if it is already set to a higher value.
db_block_size	Either this parameter must be set to 8192, or one of the following parameters have to be set: <ul style="list-style-type: none"> ■ db_8k_cache_size ■ db_16k_cache_size ■ db_32k_cache_size <p>The db_cache_size parameter specifies in bytes of the cache of standard block size buffers. Oracle Beehive will use this value for the block size of its tablespaces.</p> <p>If any of the db_nk_cachce_size parameters are set, then Oracle Beehive will use the lowest value that is equal or greater than 8192.</p> <p>Refer to "Specifying Database Block Sizes" in "Oracle Database Administrator's Guide" for more information.</p>
java_pool_size	50M (52428800)
job_queue_processes	10
processes	150
undo_retention	3600

Note: You may have to increase the values of processes and sessions if you are using Oracle RAC.

To see all parameters (in alphabetical order) along with their current values, use the following SQL*Plus command:

```
SQL> SHOW PARAMETERS
```

You may display parameters that contain a specified string. For example, to display all parameters having `COMPATIBLE` in their names, use the following command:

```
SQL> SHOW PARAMETERS COMPATIBLE
```

Set initialization parameters with the `ALTER SYSTEM` command. For example, to set the `COMPATIBLE` parameter with the value `10.2.0.3.0`, use the following command:

```
SQL> ALTER SYSTEM SET COMPATIBLE = '10.2.0.3.0' SCOPE = SPFILE;
```

Note: After setting initialization parameters, restart the database.

Logging Settings

The archive log mode of the database must be `ARCHIVELOG`, which enables the archiving of the redo logs. Oracle Beehive uses Change Data Capture (CDC), which requires that the archive log mode be turned on. Use the following command to determine the database's archive log mode:

```
SQL> SELECT LOG_MODE FROM SYS.V$DATABASE;
```

To set the archive log mode to `ARCHIVELOG`, use the following commands:

```
SQL> shutdown immediate
SQL> startup mount exclusive
SQL> alter database archivelog;
SQL> alter database open;
```

Creating a Database

You may use Database Configuration Assistant (DBCA) to create and configure a database. For more information about this tool, refer to "Using DBCA to Create and Configure a Database" in Chapter 2, "Installing Oracle and Building the Database" in *Oracle Database 2 Day DBA*.

Tip: When creating a database for Oracle Beehive with DCBA, you may choose any template. With DCBA, you may also turn on the archive log mode (by selecting the option **Enable Archiving**), set the character set to `AL32UTF8`, and set initialization parameters listed in "[Initialization Parameters](#)".

Using Oracle Real Application Clusters Database

Refer to "[Configuring and Installing Oracle Beehive for Oracle RAC](#)" for pre-installation and post-installation steps required for Oracle Beehive deployments that use Oracle Real Application Clusters (RAC) databases.

Customizing Oracle Beehive Tablespaces Layout

You may customize the layout of Oracle Beehive tablespaces with the script `<Oracle home>/beehive/db/framework/beehive_custom_ts.sql`. You may change tablespace names, initial size of the datafile, block size, and datafile autoextend options.

By default, Oracle Beehive uses eleven tablespaces. The `beehive_custom_ts.sql` script maps each tablespace to a placeholder. The Oracle Beehive Install Wizard uses these placeholders (instead of the real tablespace name) for configuration.

For example, in the `beehive_custom_ts.sql` script, the tablespace `BEE_DATA` is mapped to the placeholder `ts_data`:

```
REM Tables containing transactional data
DEFINE ts_data          = BEE_DATA
DEFINE ts_data_path     = ''
DEFINE ts_data_sz       = 512M
DEFINE ts_data_bsz      = 8192
DEFINE ts_data_ae       = ON
DEFINE ts_data_ae_nx    = 128M
```

You must specify the mapping between the placeholders, such as `ts_data` and `ts_index`, and the corresponding tablespaces. If one tablespace is to be shared across multiple placeholders, adjust the tablespace size accordingly.

This script creates a special view named `bee_tablespaces`, which then is used by the install scripts. You may drop the view after a successful installation. Since the tablespace mapping information can be reused for subsequent product reinstallations, the installation script does not drop it.

The script also produces an output file with the actual `CREATE TABLESPACE` commands. The output file name is `create_beehive_tablespaces.sql`

After making your changes to the script, run it with SYS privileges.

Using Raw Storage

If you want to use a database that uses raw storage, customize the script `<Oracle home>/beehive/db/framework/beehive_custom_ts.sql`. Refer to ["Customizing Oracle Beehive Tablespaces Layout"](#) for more information about this script.

To use a raw device (which you have created and properly configured) for a particular tablespace, specify the full path of its datafile in the appropriate variable. For example, if you want to use a raw device for the `BEE_DATA` tablespace, specify the full path of its datafile in the `ts_data_path` variable.

By default, Oracle Beehive uses eleven tablespaces. The `beehive_custom_ts.sql` script maps each tablespace to a placeholder. The Oracle Beehive Install Wizard uses these placeholders (instead of the real tablespace name) for configuration.

To use a raw device (which you have created and properly configured) for a particular tablespace, specify the full path of its datafile in the appropriate variable. For example, if you want to use a raw device for the `BEE_DATA` tablespace, specify the full path of its datafile in the `ts_data_path` variable.

In addition, set the variable `APPEND_FILE_NAME` to `N`.

After making your changes to the script, run it with SYS privileges.

Oracle Beehive Install Wizard Checklist

This module lists the information the Oracle Beehive Install Wizard requests during the installation process.

The Oracle Beehive Install Wizard asks you to choose from one of the following installation scenarios:

- [Oracle Beehive \(Standard Installation\)](#)
- [Oracle Beehive Provisioning Application](#)
- [Oracle Beehive for DMZ](#)
- [Oracle Beehive Integration for Zimbra](#)
- [Oracle Beehive Administration Console](#) (install this product with the Oracle Beehive Administration Console Install Wizard)

Refer to "[Oracle Beehive Install Wizard](#)" for detailed information about the Oracle Beehive Install Wizard.

Refer to "[Oracle Beehive Administration Console Installation Help](#)" for detailed information about the Oracle Beehive Administration Console Install Wizard.

Oracle Beehive (Standard Installation)

- [New Inventory Location](#): Oracle inventory directory path
The Oracle Beehive Install Wizard will ask you for the Oracle inventory directory path if Oracle Beehive is the first Oracle product you are installing on your system.
- [Specify Home Location](#): Full path of the location you want to install Oracle Beehive.
- [Database Information](#)
 - [Host and Port](#)
 - [Service Name](#)
 - [Admin User Name](#)
 - [Admin User Password](#)
- [Enterprise Definition Information](#)
 - [Enterprise](#)
 - [Organization](#)
 - [Instance Name](#)

- Site
- Schema Password or Existing Schema Password
- beeadmin Password

Note: The Oracle Beehive Install Wizard will ask for all this information if you select the [Install and Configure](#) option.

If you choose the [Install Only](#) option, the Oracle Beehive Install Wizard will only ask you for the Oracle home location.

Oracle Beehive Provisioning Application

- [New Inventory Location](#)
- [Specify Home Location](#): Full path of Oracle Management Service Oracle home
- [Repository Database Administrator Password](#): Password for the SYS account of the database that the Oracle Management Service uses

Oracle Beehive for DMZ

- [New Inventory Location](#)
- [Specify Home Location](#): Full path of the location where you want to install Oracle Beehive and configure as a DMZ

Oracle Beehive Integration for Zimbra

Oracle Beehive Integration for Zimbra is available for Oracle Beehive Release 1 (1.3) and later.

- [Specify Home Location](#): Full path of the location where you want to install Oracle Beehive Integration for Zimbra
- [Database Information](#)
 - [Host and Port](#)
 - [Service Name](#)
 - [Admin User Name](#)
 - [Admin User Password](#)
- [Existing Enterprise Information](#)
 - [Enterprise](#)
 - [Site](#)
 - [Instance](#)
- [Existing Schema Password](#)

Oracle Beehive Administration Console

Oracle Beehive Administration Console is available for Oracle Beehive Release 1 (1.3) and later.

- [Specify Home Location](#): Full path of the location where you want to install Oracle Beehive Administration Console
- [Database Information](#)
 - [Host and Port](#)
 - [Service Name](#)
 - [Schema User Password](#)

Oracle Beehive Install Wizard

The following sections describe the Oracle Beehive Install Wizard screens that appear during installation of Oracle Beehive:

- [Product Selection Sequence of Screens](#)
- [Oracle Beehive \(Standard Installation\) Sequence of Screens](#)
- [Oracle Beehive Provisioning Application Sequence of Screens](#)
- [Oracle Beehive for DMZ Sequence of Screens](#)
- [Oracle Beehive Integration for Zimbra Sequence of Screens](#)

Product Selection Sequence of Screens

The following screens appear prior to product selection:

- [Welcome](#)
- [New Inventory Location](#)
- [Select Product Type](#)

Once you select a product type, you will proceed to one of the following sequence of screens:

- [Oracle Beehive \(Standard Installation\) Sequence of Screens](#)
- [Oracle Beehive Provisioning Application Sequence of Screens](#)
- [Oracle Beehive for DMZ Sequence of Screens](#)
- [Oracle Beehive Integration for Zimbra Sequence of Screens](#)

Welcome

This screen welcomes you to the Oracle Install Wizard. It will guide you through the installation and configuration of Oracle Beehive Release 1.

The following describes the buttons that appear on most of the installation screens:

- Click **Help** to view the online help.
- Click **Save** to save the inputs you have entered to a file that you can use later as a response file. You can use this response file to continue the installation at a later time or perform a silent (non-interactive) installation.

Refer to the module "[Installing Oracle Beehive in Silent Mode \(Non-Interactive\)](#)" in *Oracle Beehive Installation Guide* for detailed information about installing Oracle Beehive with a response file.

- Click **Back** to go back to the previous step.
- Click **Next** to go to the next step.
- Click **Install** to start the installation of the product.
This button is available only on the **Summary** screen.
- Click **Uninstall** to uninstall a previously installed Oracle Beehive product
- Click **Cancel** to exit from the Oracle Install Wizard at any time during the installation.

Tip: Press **F1** to open the online help section for the field where your cursor focus is on the installation screen.

New Inventory Location

If Oracle Beehive Release 1 is the first Oracle product you are installing on a computer, then this screen is displayed.

On this screen, you specify in the **Inventory Directory Path** field the location of an inventory directory (the `oraInventory` directory). The Oracle Beehive Install Wizard will use this inventory directory to keep track of all Oracle products installed on the computer.

Notes: The following notes are about the inventory directory:

- If an Oracle product was installed previously on the computer, then the Oracle Beehive Install Wizard uses the existing inventory directory. To ensure that you have write permissions on that directory, run the Oracle Beehive Install Wizard as the same operating system user who installed the existing Oracle product.
- The inventory directory location should be different from the directory path in which you are going to install Oracle Beehive Release 1.
- The Oracle Beehive Install Wizard (as well as other installation programs from other Oracle products) will store its log files in the directory `<inventory directory path>/logs`.

However, if this is the first Oracle product you are installing on your computer, the Install Wizard will store its log files in a temporary directory, such as `/tmp` on Linux and UNIX-based operating systems. When you exit the Install Wizard, the Install Wizard will move these log files to `<inventory directory path>/logs`.

On **Linux and UNIX-based platforms**, specify in the **Operating System Group** field the operating system group that has the `write` permission to the target inventory directory location.

Select Product Type

Choose how you want to install Oracle Beehive from one of the following options:

- Complete installation by using the [Oracle Beehive Release 1](#) option
- Plug-in installation by using the [Provisioning Application for Oracle Beehive Release 1](#) option

- Installation for a demilitarized zone by using the [Oracle Beehive Release 1 for DMZ](#) option
- Installation of [Oracle Beehive Integration for Zimbra](#), a Web-based client for Oracle Beehive.

The sequence of the following screens depend on the option that you select on this screen.

Oracle Beehive Release 1

Use this option to perform a complete installation of Oracle Beehive Release 1 on the specified host.

Note: One of the prerequisites for selection the Oracle Beehive Release 1 option is that you must have an existing instance of Oracle Database. This database is not installed by this procedure.

Refer to the module "[Oracle Beehive Database Requirements](#)" in the Oracle Beehive Installation Guide of your operating system for database requirements for Oracle Beehive Release 1.

See "[Oracle Beehive \(Standard Installation\) Sequence of Screens](#)" to continue installing Oracle Beehive Release 1.

Provisioning Application for Oracle Beehive Release 1

Use this option to install the Provisioning Application. This will allow you to install and configure Oracle Beehive Release 1 instances from Oracle Enterprise Manager Grid Control.

The prerequisite for installing this plug-in is that you must have a preinstalled instance of Oracle Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) or later.

See "[Oracle Beehive Provisioning Application Sequence of Screens](#)" to continue installing Oracle Beehive Release 1 Provisioning Application.

Oracle Beehive Release 1 for DMZ

Use this option to install Oracle Beehive Release 1 and configure the installation location as a demilitarized zone (DMZ).

This option uses a local, file-based configuration and does not require an existing database. In addition, the administration tool `beectl` will be disabled for this instance.

See "[Oracle Beehive for DMZ Sequence of Screens](#)" to continue installing Oracle Beehive Release 1 for DMZ.

Oracle Beehive Integration for Zimbra

Oracle Beehive Integration for Zimbra is available for Oracle Beehive Release 1 (1.3) and later.

Use this option to install Oracle Beehive Integration for Zimbra, a Web-based client for Oracle Beehive.

See "[Oracle Beehive Integration for Zimbra Sequence of Screens](#)" to continue installing Oracle Beehive Integration for Zimbra.

Oracle Beehive (Standard Installation) Sequence of Screens

The following screens appear during the installation of Oracle Beehive Release 1 (standard installation):

- [Specify Home Location](#)
- [Prerequisite Checks](#)
- [Select Installation Type](#)
- [Database Information](#)
- [Enterprise Definition Information](#)
- [Existing Enterprise Information](#)
- [Schema Password](#)
- [Existing Schema Password](#)
- [beeadmin Password](#)
- [Installation Summary](#)
- [Progress Indicator](#)
- [Configuration Assistants](#)
- [End of Installation](#)

Other Oracle Install Wizard Sequence of Screens

If you are installing Oracle Beehive Provisioning Application, refer to "[Oracle Beehive Provisioning Application Sequence of Screens](#)".

If you are installing Oracle Beehive for DMZ, refer to "[Oracle Beehive for DMZ Sequence of Screens](#)".

Specify Home Location

Enter the full path of the location where you want to install the product. Do not leave the **Location** field blank.

Note: If you selected the **Provisioning Application for Oracle Beehive Release 1** option on the [Select Product Type](#) screen, ensure that you specify the Oracle home for your Oracle Enterprise Manager Grid Control instance. This is a prerequisite for installing the provisioning plug-in.

For more information on prerequisites for installing the provisioning plug-in, refer to "[Provisioning Application for Oracle Beehive Release 1](#)". Refer to the module "[What You Should Know Before Installing Oracle Beehive](#)" in *Oracle Beehive Installation Guide* for information on prerequisites for installing Oracle Beehive Release 1.

Specifying Oracle Home Directory

Specify the directory where you want to install Oracle Beehive. You must specify a new Oracle home directory for each new installation of Oracle Beehive.

See the following guidelines for specifying the Oracle home directory on different platforms:

- [Specifying Oracle Home Directory on Linux and UNIX-Based Systems](#)
- [Specifying Oracle Home Directory on Windows Systems](#)

Specifying Oracle Home Directory on Linux and UNIX-Based Systems

On Linux and UNIX-based systems, the Oracle Beehive Install Wizard suggests a path similar to the following for the Oracle home directory:

```
/home_directory/oracle/product/version_number/identifier_n
```

- *home_directory* is the home directory of the user running the Oracle Beehive Install Wizard
- *version_number* is the version number of the product being installed
- *identifier* identifies the product installed in this Oracle home directory
- *n* is a number that ensures that this path is unique

Note: You must have write permissions in the directory that you specify, if it exists, or you must have permissions to create it.

You may also click **Browse** to choose a directory to install your product. This location is the destination directory in which the product will be installed.

Specifying Oracle Home Directory on Windows Systems

On Windows systems, the Oracle Beehive Install Wizard suggests a path similar to the following for the Oracle home directory:

```
X:\oracle\product\version_number\identifier_n
```

- *X* is the default drive
- *version_number* is the version number of the product being installed
- *identifier* identifies the product installed in this Oracle home directory
- *n* is a number that ensures that this path is unique

If the directory specified does not exist, the Oracle Beehive Install Wizard creates it. Unless you want to install the software on a different drive with sufficient free space, you can accept the default value.

You may also click **Browse** to choose a directory to install your product. This location is the destination directory in which the product will be installed.

Prerequisite Checks

This screen displays the prerequisite checks that the Oracle Beehive Install Wizard runs. It verifies that the host (where you are installing Oracle Beehive Release 1) meets all minimum requirements for installing and configuring the product type that you selected on the [Select Product Type](#) screen.

Some of the platform-independent checks that the Oracle Beehive Install Wizard performs include:

- Operating system certification (or version)
- Operating system patches and packages
- Security kernel parameters

- Memory
- Swap space
- Disk space
- In Microsoft Windows, ensuring that the Window user for Oracle Beehive installation has administrative privileges.

If an automatic check fails, fix it and click **Retry**.

Notes: If the disk space check fails and after clicking **Retry** (after fixing the low disk space issue) the check states "Not Executed," exit the Oracle Beehive Install Wizard and start it again.

In Microsoft Windows, The Windows user indicated in the **Browse and Select: File or Directory** screen must belong to the "Log on as a batch job" policy of the agent machine.

To add a user to this policy, click **Start, Programs, Administrative Tools**, and then **Local Security Policy**. In **Local Security Settings**, expand **Local Policies**, expand **Local Policies**, and then click **User Rights Assignment**. Double-click the policy **Log on as a batch job**. Click **Add User or Group** to add the user.

Tip: For details on why a check failed, select the check box against it and see the details in the description box at the bottom of the screen.

Select Installation Type

Specify the type of installation you want to perform. You can choose from:

- [Install and Configure](#)
- [Install Only](#)

Install and Configure

Select this option if you want to install and automatically start the configuration of the product.

If you select this option, the Oracle Beehive Release 1 instance is functional with minimal configuration after the installation is completed.

Install Only

Select this option to only install the product. This will only copy files to your system. Until you configure Oracle Beehive, you will not be able to perform any tasks with it nor start any Oracle Beehive managed component.

To configure the product after installation, you must run the Configuration wizard as follows:

1. Navigate to the following directory:
 - **On Linux and UNIX-based platforms:** `$ORACLE_HOME/bee hive/oobwiz`
 - **On Microsoft Windows:** `%ORACLE_HOME%\bee hive\oobwiz`
2. Run the `configWizard` script as follows:
 - **On Linux and UNIX-based platforms,** enter the following command:

```
./configWizard
```

- **On Microsoft Windows**, run the `configWizard.bat` script.

Database Information

Specify the following details of an existing Oracle Database instance that will be used by this installation:

- [Host and Port](#)
- [Service Name](#)
- [Admin User Name](#)
- [Admin User Password](#)

Notes: The database that you specify on this screen must use the Unicode Standard UTF-8 AL32UTF8 character set.

To determine the character set of an existing database, perform the following steps:

1. Log in to the SQL*Plus console of the target Oracle database.
2. Run the following query:

```
select value from nls_database_parameters where parameter='NLS_
CHARACTERSET';
```

If the character set of the Oracle database is not Unicode Standard UTF-8 AL32UTF8, then you must create a new Oracle database. You may use Database Configuration Assistant (DBCA) to do this.

Refer to the module "[Oracle Beehive Database Requirements](#)" in the Oracle Beehive Installation Guide of your operating system for database requirements for Oracle Beehive Release 1.

Host and Port

Host

Specify the fully qualified domain name (FQDN) of the host where the target database is running.

For example, if the host name is `foo`, its domain is `abc.com`, and the database listener port is 1521, then enter the following:

```
foo.abc.com:1521
```

The default port on which the database listener listens for connection requests is 1521.

To determine this port number, run the command `<Database home>/bin/lsnrctl status`. (Ensure that the environment variable `ORACLE_HOME` is set to the directory in which Oracle Database is installed and `ORACLE_SID` is set to the system identifier of your database.)

Service Name

Enter the service name for the Oracle Database. This is the same as the global database name, and must be unique across all databases.

A database is identified by its *global database name*. The global database name is comprised of two parts:

`database_name.database_domain`

For example:

`sales.us.yourcompany.com`

Admin User Name

Specify the administrative user name (that can perform database administration tasks) for the database instance. Typically, this user is the `SYS` account.

For more information about the `SYS` account, refer to *Oracle Database Administrator's Guide*.

Admin User Password

Enter the password for the Oracle database administrative user account that you specified in the **Admin User Name** field.

After clicking **Next**, the Oracle Beehive Install Wizard verifies the connection to the Oracle Database Server, initialization parameters, and other requirements. If an error occurs, then you must correct the problem before the installation process can continue.

Enterprise Definition Information

Specify the following enterprise site information that identifies this installation of Oracle Beehive Release 1:

- [Enterprise](#)
- [Organization](#)
- [Instance Name](#)
- [Site](#)

Enterprise

Specify the name for the new enterprise you want to create. An enterprise is the top-level container for all users and objects. An enterprise may contain any number of organizations and workspaces.

The enterprise name is typically a company name, such as `Oracle`.

The enterprise name may only contain only letters, numbers, spaces, the period (`.`), the hyphen (`-`) and the underscore (`_`). The name cannot be greater than 127 bytes. The name cannot consist of only one or two periods (`.` or `..`).

In Oracle Beehive Release 1, an Oracle Beehive deployment may have only one enterprise.

Organization

Specify the name of the organization to be created and added to the enterprise you specified previously. An organization is a logical grouping of users, groups, workspaces, and resources at a level lower than the enterprise level. Later, you may add any number of organizations to the enterprise you specified previously.

The organization is typically a sub-level or department under the enterprise, such as `Sales` or `Accounting`.

The enterprise name may only contain only letters, numbers, spaces, the period (`.`), the hyphen (`-`) and the underscore (`_`). The name cannot be greater than 127 bytes. The name cannot consist of only one or two periods (`.` or `..`).

Instance Name

Specify the name of the instance for this Oracle Beehive installation. The instance name is used during server configuration.

The full name of an Oracle Beehive instance consists of the instance, host, and domain names. For example, if you name your instance `myinstance1`, your host is `myhost`, and your domain is `example.com`, the full name of your instance will be `myinstance1.myhost.example.com`.

The instance name may only contain ASCII letters (a-z, A-Z), numbers, and the underscore (`_`). The name cannot be greater than 63 characters. The name cannot be `tempinst`.

Notes: Only enter the instance name; do not specify the host and domain name. Oracle Beehive will append the host and domain name to the instance name. For example, you would specify `myinstance1`, not `myinstance1.myhost.example.com`.

All Oracle Beehive Release 1 instances must have a unique name.

Site

Specify the name of the site. A site consists of all Oracle Beehive services and service instances and the database they use.

A site is typically a reference to a physical location, such as `Headquarters` or `Montreal`.

The site name may only contain ASCII letters (a-z, A-Z), numbers, and the underscore (`_`). The name cannot be greater than 19 characters.

In Oracle Beehive Release 1, an Oracle Beehive deployment may have only one site.

Existing Enterprise Information

Specify the name of the instance for this Oracle Beehive installation. The instance name is used during server configuration.

Note: This screen will only appear if you are installing Oracle Beehive Release 1 against a database that is already configured for an existing Oracle Beehive instance.

The Oracle Beehive Install Wizard will use the enterprise and site of the existing Oracle Beehive instance; the enterprise and site name displayed on this screen are those of this Oracle Beehive instance.

Schema Password

Specify and confirm the password for the new database schema that will be created and used by the instance of Oracle Beehive Release 1.

Password Restrictions

The schema password must have the following characteristics:

- Be between 5 and 30 characters long.
- Only contain alpha-numeric characters (a-z, A-Z, and 0-9) or the underscore (`_`); the password cannot contain any other punctuation mark character.

- Contain at least one alphabetic character and at least one numeric character
- Begin with an alphabetic character; the password cannot begin with a number or the underscore (_).

Password Recommendations

- Passwords should not be simple or obvious words, such as welcome, account, database, or user.
- Passwords should not be the same as the user name.
- Passwords should not be Oracle reserved words such as ACCESS, AUTHORIZATION, UID, or VARCHAR,

Existing Schema Password

Enter the password for the Oracle database schema that the Oracle Beehive Install Wizard has detected. This schema is used by the Oracle Beehive Release 1 instance.

After clicking **Next**, the Oracle Beehive Install Wizard verifies the schema password. If an error occurs, then you must correct the password before the installation process can continue.

beeadmin Password

Specify and confirm the password for the beeadmin user. The oc4jadmin and bpeladmin users will have the same password.

The oc4jadmin user will be exposed if you set up monitoring with Oracle Enterprise Manager Grid Control.

Use the bpeladmin user to log into Oracle BPEL Process Manager. Do not use Oracle BPEL Console to deploy or modify workflows.

Note: This screen will not appear if you are installing Oracle Beehive against a database that is already configured for an existing Oracle Beehive instance.

Password Restrictions

The password of the beeadmin user must have the following characteristics:

- Be between 8 and 30 characters long.
- Only contain alpha-numeric characters (a-z, A-Z, and 0-9) or the underscore (_); the password cannot contain any other punctuation mark character.
- Contain at least one uppercase and one lowercase alphabetic character.
- Contain at least one numeric character.
- Begin with an alphabetic character; the password cannot begin with a number or the underscore (_).

Note: When entering your password, check that the state of your Caps Lock key is what you want it to be. Passwords are case-sensitive.

Password Recommendations

- Passwords should not be simple or obvious words, such as welcome, account, database, or user.
- Passwords should not be the same as the user name.
- Passwords should not be Oracle reserved words such as ACCESS, AUTHORIZATION, UID, or VARCHAR,

Installation Summary

The Installation Summary screen presents a summary of the options you have selected so far. Depending on the installation, the information may include the following:

- **Space Requirements:** The disk space required to install the product. The space requirement appears in a different color if the disk space available is less than what is required.

Notes: The space requirements shown on this screen do not include the space required in the target database.

The space requirements shown on this screen also might differ slightly from the actual space the component occupies on the installation disk. This is because of the differences in the disk configurations. For example, the space required on one hard drive might differ from the space required on another hard drive that uses different block size or that is managed by a different file system.

- **Product Type:** A predefined component set that automatically selects which components and dependency groups to install. To change the product type, go back to the [Select Product Type](#) screen.
- **Install Type:** The option to only install Oracle Beehive Release 1 or to install and configure Oracle Beehive Release 1. To change the install type, go to the [Select Installation Type](#) screen.
- **Inventory Location:** The location of the `oraInventory` directory where all the installation information about your product will be stored. To change this location, go back to the [New Inventory Location](#) screen.
- **Interview Details:** The details of your inputs for the other screens, such as:
 - **Home Details:** The location of the Oracle home where you want to install your product. To change the Oracle home, go back to the [Specify Home Location](#) screen.
 - **Database Information:** The host name, port, service name, administrator user name, and password of the existing Oracle Database instance that will be used by this installation. To change the details, go back to the [Database Information](#) screen.
 - **Enterprise Definition Information:** The detailed enterprise site information that identifies this installation. To change the details, go back to the [Enterprise Definition Information](#) screen.
 - **Schema Password:** The masked password for the Oracle database schema that the Oracle Beehive Install Wizard detects and is used by this Oracle Beehive instance.

- **beadmin Password:** The masked password for the `beadmin` user, which is the administrative user for Oracle Beehive Release 1 instances. To change the details, go back to the [beadmin Password](#) screen.

When you have reviewed your inputs, click **Install** to continue.

Progress Indicator

Use this screen to monitor the progress of your installation. Click **Show Details** to see the details in the description box that appears.

Configuration Assistants

This screen displays the configuration assistants that the Oracle Beehive Install Wizard runs to configure the product type that you selected on the [Select Product Type](#) screen.

The Oracle Beehive Install Wizard performs the following configuration assistants:

- Oracle Beehive Product Preparation Assistant
- Oracle Beehive Initialization Assistant
- Oracle Beehive Database Configuration Assistant - SOA schemas
- Base Platform Support Configuration Assistant
- Disabling OC4J Home Instance
- Oracle Beehive Database Configuration Assistant - Beehive schemas
- Oracle Beehive Framework Core Configuration Assistant

If any configuration is flagged as Failed, then you must manually run the assistant to configure it. You do so by selecting the check box against it and clicking **Retry**.

You may also use the Oracle Beehive Config Wizard to complete all pending and failed configuration assistants at a later time. The full path of the command is `<Oracle home>/beehive/oobwiz/configWizard`.

For more information about the Oracle Beehive Config Wizard, refer to the module "[Oracle Beehive Install and Config Wizard Command-Line Options](#)" in *Oracle Beehive Installation Guide*.

Tip: For details on why a configuration failed, select the check box against it and see the details in the description box at the bottom of the screen. Also, review the log files in `<Oracle inventory directory>/logs` whose names start with `installActions`.

End of Installation

Click **Finish** to complete the installation process and close the Oracle Beehive Install Wizard.

Oracle Beehive Provisioning Application Sequence of Screens

The following screens appear during the installation of Oracle Beehive Release 1 Provisioning Application:

- [Specify Home Location](#)
- [Repository Database Administrator Password](#)
- [Installation Summary](#)

- [Progress Indicator](#)
- [Configuration Assistants](#)
- [End of Installation](#)

Other Oracle Beehive Install Wizard Sequence of Screens

If you are installing Oracle Beehive (standard installation), refer to "[Oracle Beehive \(Standard Installation\) Sequence of Screens](#)"

If you are installing Oracle Beehive for DMZ, refer to "[Oracle Beehive for DMZ Sequence of Screens](#)".

Specify Home Location

In the **Location** field, specify the full path of Oracle Management Service Oracle home in which you want to install Oracle Beehive Provisioning Application.

Oracle Management Service is part of Oracle Enterprise Manager Grid Control.

You may not leave the **Location** field blank.

Typically, the name of this directory is oms10g.

Repository Database Administrator Password

In the **SYS Password** field, specify the password for the *SYS* account of the repository database that Oracle Management Service uses.

This repository database will be used by Oracle Beehive Provisioning Application to store required information.

The screen displays the details of the Oracle Enterprise Manager database that is used as the repository for the Oracle Beehive Release 1 provisioning plug-in that you are installing. It displays the following information:

- Host name where the Oracle database is installed
- Port on which it listens to the incoming requests
- System identifier (SID) that defines the name of the Oracle database instance

The Oracle Beehive Install Wizard verifies the password. If you specified an incorrect password, an error occurs. In this case, you must correct the password before the installation process can continue.

Installation Summary

The Installation Summary screen presents a summary of the options you have selected so far. Depending on the installation, the information may include the following interview details:

- **Home Detail:** The location of the Oracle home where you want to install your product. To change the Oracle home, go back to the [Specify Home Location](#) screen.
- **Repository Administrator Password:** The masked password for the *SYS* user, which is the administrative user for the database that this Oracle Beehive Release 1 instance uses as a repository.

When you have reviewed your inputs, click **Install** to continue.

Progress Indicator

Use this screen to monitor the progress of your installation. Click **Show Details** to see the details in the description box that appears.

Configuration Assistants

This screen displays the configuration assistants that the Oracle Beehive Install Wizard runs to configure Oracle Beehive Provisioning Application.

The Oracle Beehive Install Wizard performs the following configuration assistants:

- Repository Upgrade
- OMS Redeployment
- Deploy Provision Application
- Start Oracle Management Server

If any configuration is flagged as Failed, then you must manually run the assistant to configure it. You do so by selecting the check box against it and clicking **Retry**.

Tip: For details on why a configuration failed, select the check box against it and see the details in the description box at the bottom of the screen. Also, review the log files in `<Oracle inventory directory>/logs` whose names start with `installActions`.

End of Installation

The End of Installation screen indicates if the installation of Oracle Beehive Provisioning Application was successful.

Click **Finish** to complete the installation process and close the Oracle Beehive Install Wizard.

Refer to "[Installing Oracle Beehive with Provisioning Application](#)" to install Oracle Beehive with Oracle Beehive Provisioning Application.

Oracle Beehive for DMZ Sequence of Screens

The following screens appear during the installation of Oracle Beehive Release 1 for DMZ:

- [Specify Home Location](#)
- [Prerequisite Checks](#)
- [Installation Summary](#)
- [Progress Indicator](#)
- [Configuration Assistants](#)
- [End of Installation](#)

Other Oracle Beehive Install Wizard Sequence of Screens

If you are installing Oracle Beehive (standard installation), refer to "[Oracle Beehive \(Standard Installation\) Sequence of Screens](#)"

If you are installing Oracle Beehive Provisioning Application, refer to "[Oracle Beehive Provisioning Application Sequence of Screens](#)".

Specify Home Location

Enter the complete path for the location where you want to install Oracle Beehive Release 1 and configure as a DMZ.

Prerequisite Checks

This screen displays the prerequisite checks that the Oracle Beehive Install Wizard runs. It verifies that the host (where you are installing Oracle Beehive Release 1 for DMZ) meets all minimum requirements for installing and configuring the product type that you selected on the [Select Product Type](#) screen.

Some of the platform-independent checks that the Oracle Beehive Install Wizard performs include:

- Operating system certification (or version)
- Operating system patches and packages
- Security kernel parameters
- Memory
- Swap space
- Disk space

Tip: For details on why a check failed, select the check box against it and see the details in the description box at the bottom of the screen.

Installation Summary

The Installation Summary screen presents a summary of the options you have selected so far. The information includes the following:

- **Space Requirements:** The disk space required to install the product. The space requirement appears in a different color if the disk space available is less than what is required.

Note: The space requirements shown on this screen might differ slightly from the actual space the component occupies on the installation disk. This is because of the differences in the disk configurations. For example, the space required on one hard drive might differ from the space required on another hard drive that uses different block size or that is managed by a different file system.

- **Inventory Location:** The location of the `oraInventory` directory where all the installation information about your product will be stored. To change this location, go back to the [New Inventory Location](#) screen.
- **Interview Details:** The details of your inputs for the other screens:
 - **Home Details:** The location of the Oracle home where you want to install Oracle Beehive Release 1 for DMZ. To change the Oracle home, go back to the [Specify Home Location](#) screen.

Progress Indicator

Use this screen to monitor the progress of your installation. Click **Show Details** to see the details in the description box that appears.

Configuration Assistants

This screen displays the configuration assistants that the Oracle Beehive Install Wizard runs to configure Oracle Beehive Release 1 for DMZ.

The Oracle Beehive Install Wizard performs the following configuration assistants:

- Oracle Beehive for DMZ Product Preparation
- Oracle Beehive for DMZ Initialization
- Disabling OC4J Instances
- BTI (Oracle Beehive Transport Infrastructure) Configuration

If any configuration is flagged as Failed, then you must manually run the assistant to configure it. You do so by selecting the check box against it and clicking **Retry**.

You may also use the Oracle Beehive Config Wizard to complete all pending and failed configuration assistants at a later time. The full path of the command is `<Oracle home>/beehive/oobwiz/configWizard`.

For more information about the Oracle Beehive Config Wizard, refer to the module "[Oracle Beehive Install and Config Wizard Command-Line Options](#)" in *Oracle Beehive Installation Guide*.

Tip: For details on why a configuration failed, select the check box against it and see the details in the description box at the bottom of the screen. Also, review the log files in `<Oracle inventory directory>/logs` whose names start with `installActions`.

End of Installation

The End of Installation screen indicates if the installation of Oracle Beehive Release 1 for DMZ was successful.

Click **Finish** to complete the installation process and close the Oracle Beehive Install Wizard.

Refer to "[Configuring Oracle Beehive Demilitarized Zone Instances](#)" to configure the Oracle Beehive DMZ instance you have just installed.

Oracle Beehive Integration for Zimbra Sequence of Screens

Oracle Beehive Integration for Zimbra is available for Oracle Beehive Release 1 (1.3) and later.

The following screens appear during the installation of Oracle Beehive Integration for Zimbra:

- [Specify Home Location](#)
- [Prerequisite Checks](#)
- [Select Installation Type](#)
- [Database Information](#)
- [Existing Enterprise Information](#)
- [Existing Schema Password](#)
- [Installation Summary](#)
- [Progress Indicator](#)

- [Configuration Assistants](#)
- [End of Installation](#)

Specify Home Location

Enter the full path of the location where you want to install Oracle Beehive Integration for Zimbra. Do not leave the **Location** field blank.

Prerequisite Checks

This screen displays the prerequisite checks that the Oracle Beehive Install Wizard runs. It verifies that the host (where you are installing Oracle Beehive Integration for Zimbra) meets all minimum requirements for installing and configuring the product type that you selected on the [Select Product Type](#) screen.

Some of the platform-independent checks that the Oracle Beehive Install Wizard performs include:

- Operating system certification (or version)
- Operating system patches and packages
- Security kernel parameters
- Memory
- Swap space
- Disk space

If an automatic check fails, fix it and click **Retry**.

Notes: If the disk space check fails and after clicking **Retry** (after fixing the low disk space issue) the check states "Not Executed," exit the Oracle Beehive Install Wizard and start it again.

Tip: For details on why a check failed, select the check box against it and see the details in the description box at the bottom of the screen.

Select Installation Type

Specify the type of installation you want to perform. You can choose from:

- [Install and Configure](#)
- [Install Only](#)

Install and Configure

Select this option if you want to install and automatically start the configuration of the product.

If you select this option, Oracle Beehive Integration for Zimbra is functional with minimal configuration after the installation is completed.

Install Only

Select this option to only install the product. This will only copy files to your system. Until you configure Oracle Beehive Integration for Zimbra, you will not be able to perform any tasks with it.

To configure the product after installation, you must run the Configuration wizard as follows:

1. Navigate to the following directory:
 - **On Linux and UNIX-based platforms:** `$ORACLE_HOME/bee hive/oobwiz`
 - **On Microsoft Windows:** `%ORACLE_HOME%\bee hive\oobwiz`
2. Run the `configWizard` script as follows:
 - **On Linux and UNIX-based platforms**, enter the following command:
`./configWizard`
 - **On Microsoft Windows**, run the `configWizard.bat` script.

Database Information

Specify the following details of an existing Oracle Database instance that will be used by this installation:

- [Host and Port](#)
- [Service Name](#)
- [Admin User Name](#)
- [Admin User Password](#)

Note: The database that you specify on this screen must use the Unicode Standard UTF-8 AL32UTF8 character set.

To determine the character set of an existing database, perform the following steps:

1. Log in to the SQL*Plus console of the target Oracle database.
2. Run the following query:

```
select value from nls_database_parameters where parameter='NLS_CHARACTERSET';
```

If the character set of the Oracle database is not Unicode Standard UTF-8 AL32UTF8, then you must create a new Oracle database. You may use Database Configuration Assistant (DBCA) to do this.

Host and Port

Host

Specify the fully qualified domain name (FQDN) of the host where the target database is running.

For example, if the host name is `foo`, its domain is `abc.com`, and the database listener port is `1521`, then you must enter the following:

```
foo.abc.com:1521
```

In case of an Oracle Real Application Cluster (Oracle RAC) database, use the following format:

```
virtual_host1:virtual_port1^host2:virtual_port2^.....
```

Port

Specify the port on which Oracle database is listening.

You can determine the Oracle database Listener port value from the following file in the Database Oracle home:

- **On Linux and UNIX-based platforms:**

```
$ORACLE_HOME/network/admin/listener.ora
```

- **On Microsoft Windows:**

```
%ORACLE_HOME%\network\admin\listener.ora
```

If you are using Oracle Enterprise Manager, then to determine the port on which Oracle database is listening, click the **Ports** link in Oracle Enterprise Manager 10g Database Control for the instance where Oracle database is installed.

The default port on which the database listener listens for connection requests is 1521.

To check if a port is being used, you can use the `netstat` command to show the used port as follows:

- **On Linux and UNIX-based platforms:**

```
prompt> netstat -an | grep port_num
```

In the preceding syntax, `port_num` refers to the port number you want to check.

- **On Microsoft Windows:**

```
Drive_Letter:\> netstat -an | find "port_num"
```

In the preceding syntax, `port_num` refers to the port number you want to check.

Note that you need quotation marks around the port number.

Service Name

Enter the service name for the Oracle Database. This is the same as the global database name, and must be unique across all databases.

A database is identified by its *global database name*. The global database name is comprised of two parts:

```
database_name.database_domain
```

For example:

```
sales.us.yourcompany.com
```

Admin User Name

Specify the administrative user name (that can perform database administration tasks) for the database instance. Typically, this user is the `SYS` account.

For more information about the `SYS` account, refer to *Oracle Database Administrator's Guide*.

Admin User Password

Enter the password for the Oracle database administrative user account that you specified in the **Admin User Name** field.

After clicking **Next**, the Oracle Beehive Install Wizard verifies the connection to the Oracle Database Server, initialization parameters, and other requirements. If an error occurs, then you must correct the problem before the installation process can continue.

Existing Enterprise Information

Specify the name of the Oracle Beehive instance for this Oracle Beehive Integration for Zimbra installation. The instance name is used during server configuration.

Enterprise

Select the enterprise with which you want to configure Oracle Beehive Integration for Zimbra.

Site

This is a non-editable field displaying the configured site based on the enterprise chosen.

Instance

Specify a unique Oracle Beehive instance name for this Oracle Beehive Integration for Zimbra instance.

The full name of an Oracle Beehive instance consists of the instance, host, and domain name. For example, if you name your instance `myinstance1`, your host is `myhost`, and your domain is `example.com`, the full name of your instance will be `myinstance1.myhost.example.com`.

The instance name may only contain ASCII letters (a-z, A-Z), numbers, and the underscore (`_`). The name cannot be greater than 63 characters. The name cannot be `tempinst`.

Notes: Only enter the instance name; do not specify the host and domain name. Oracle Beehive will append the host and domain name to the instance name. For example, you would specify `myinstance1`, not `myinstance1.myhost.example.com`.

All Oracle Beehive instances must have a unique name.

Existing Schema Password

Enter the password for the Oracle database schema that the Oracle Beehive Install Wizard has detected. This schema is used by Oracle Beehive Integration for Zimbra.

After clicking **Next**, the Oracle Beehive Install Wizard verifies the schema password. If an error occurs, then you must correct the password before the installation process can continue.

Installation Summary

The Installation Summary screen presents a summary of the options you have selected so far. Depending on the installation, the information may include the following:

- **Space Requirements:** The disk space required to install the product. The space requirement appears in a different color if the disk space available is less than what is required.

Notes: The space requirements shown on this screen do not include the space required in the target database.

The space requirements shown on this screen also might differ slightly from the actual space the component occupies on the installation disk. This is because of the differences in the disk configurations. For example, the space required on one hard drive might differ from the space required on another hard drive that uses different block size or that is managed by a different file system.

- **Product Type:** A predefined component set that automatically selects which components and dependency groups to install. To change the product type, go back to the [Select Product Type](#) screen.
- **Install Type:** The option to only install Oracle Beehive Integration for Zimbra or to install and configure Oracle Beehive Integration for Zimbra. To change the install type, go to the [Select Installation Type](#) screen.
- **Inventory Location:** The location of the `oraInventory` directory where all the installation information about your product will be stored. To change this location, go back to the [New Inventory Location](#) screen.
- **Interview Details:** The details of your inputs for the other screens, such as:
 - **Home Details:** The location of the Oracle home where you want to install the product. To change the Oracle home, go back to the [Specify Home Location](#) screen.
 - **Database Information:** The host name, port, service name, administrator user name, and password of the existing Oracle Database instance that will be used by this installation. To change the details, go back to the [Database Information](#) screen.
 - **Enterprise Definition Information:** The detailed enterprise site information that identifies this installation. To change the details, go back to the [Existing Enterprise Information](#) screen.
 - **Schema Password:** The masked password for the Oracle database schema that the Oracle Beehive Install Wizard detects and is used by this Oracle Beehive instance.

When you have reviewed your inputs, click **Install** to continue.

Progress Indicator

Use this screen to monitor the progress of your installation. Click **Show Details** to see the details in the description box that appears.

Configuration Assistants

This screen displays the configuration assistants that the Oracle Beehive Install Wizard runs to configure the product type that you selected on the [Select Product Type](#) screen.

If any configuration is flagged as Failed, then you must manually run the assistant to configure it. You do so by selecting the check box against it and clicking **Retry**.

You may also use the Oracle Beehive Config Wizard to complete all pending and failed configuration assistants at a later time. The full path of the command is `<Oracle home>/beehive/oobwiz/configWizard`.

For more information about the Oracle Beehive Config Wizard, refer to the module "Oracle Beehive Install and Config Wizard Command Line Options" in *Oracle Beehive Installation Guide*.

Tip: For details on why a configuration failed, select the check box against it and see the details in the description box at the bottom of the screen.

End of Installation

Click **Finish** to complete the installation process and close the Oracle Beehive Install Wizard.

Oracle Beehive Administration Console Installation Help

Oracle Beehive Administration Console is available for Oracle Beehive Release 1 (1.3) and later.

Run `runInstaller` (UNIX-based operating systems) or `setup.exe` (Microsoft Windows) from the Oracle Beehive Administration Console installation media to start the Install Wizard for this product.

Refer to "[Starting the Oracle Beehive Administration Console Uninstall Wizard](#)" to uninstall Oracle Beehive Administration Console.

Note: Oracle Beehive Administration Console version 1.3.1.0.0 is only supported by Oracle Beehive version 1.3.1.0.0.

Oracle Beehive Administration Console Sequence of Screens

The following screens appear during the installation of Oracle Beehive Administration Console:

- [Welcome](#)
- [New Inventory Location](#)
- [Specify Home Location](#)
- [Prerequisite Checks](#)
- [Database Information](#)
- [Installation Summary](#)
- [Installation Process](#)
- [Configuration Assistants](#)
- [End of Installation](#)

Welcome

This screen welcomes you to the Oracle Install Wizard. It will guide you through the installation and configuration of Oracle Beehive Administration Console.

The following describes the buttons that appear on most of the installation screens:

- Click **Help** to view the online help.

- Click **Save** to save the inputs you have entered to a file that you can use later as a response file. You can use this response file to continue the installation at a later time.
- Click **Back** to go back to the previous step.
- Click **Next** to go to the next step.
- Click **Install** to start the installation of the product.
This button is available only on the **Summary** screen.
- Click **Uninstall** to uninstall a previously installed Oracle Beehive product
- Click **Cancel** to exit from the Oracle Install Wizard at any time during the installation.

Tip: Press **F1** to open the online help section for the field where your cursor focus is on the installation screen.

New Inventory Location

If Oracle Beehive Administration Console is the first Oracle product you are installing on a computer, then this screen is displayed.

On this screen, you specify in the **Inventory Directory Path** field the location of an inventory directory (the `oraInventory` directory). The Install Wizard will use this inventory directory to keep track of all Oracle products installed on the computer.

Notes: The following notes are about the inventory directory:

- If an Oracle product was installed previously on the computer, then the Install Wizard uses the existing inventory directory. To ensure that you have write permissions on that directory, run the Install Wizard as the same operating system user who installed the existing Oracle product.
- The inventory directory location should be different from the directory path in which you are going to install Oracle Beehive Administration Console.
- The Install Wizard (as well as other installation programs from other Oracle products) will store its log files in the directory `<inventory directory path>/logs`.

However, if this is the first Oracle product you are installing on your computer, the Install Wizard will store its log files in a temporary directory, such as `/tmp` on Linux and UNIX-based operating systems. When you exit the Install Wizard, the Install Wizard will move these log files to `<inventory directory path>/logs`.

On **Linux and UNIX-based platforms**, specify in the **Operating System Group** field the operating system group that has the `write` permission to the target inventory directory location.

Specify Home Location

Enter the full path of the location where you want to install the product. Do not leave the **Location** field blank. The location you provide must not already be present in the Oracle inventory and must be empty.

Note: You may not install Oracle Beehive Administration Console into an existing Oracle home.

Prerequisite Checks

This screen displays the prerequisite checks that the Install Wizard runs. It verifies that the host (where you are installing Oracle Beehive Administration Console) meets all minimum requirements.

Some of the platform-independent checks that the Oracle Beehive Install Wizard performs include:

- Operating system certification (or version)
- Operating system patches and packages
- Security kernel parameters
- Memory
- Swap space
- Disk space
- In Microsoft Windows, ensuring that the Window user for Oracle Beehive installation has administrative privileges.

If an automatic check fails, fix it and click **Retry**.

Notes: If the disk space check fails and after clicking **Retry** (after fixing the low disk space issue) the check states "Not Executed," exit the Install Wizard and start it again.

In Microsoft Windows, The Windows user indicated in the **Browse and Select: File or Directory** screen must belong to the "Log on as a batch job" policy of the agent machine.

To add a user to this policy, click **Start, Programs, Administrative Tools**, and then **Local Security Policy**. In **Local Security Settings**, expand **Local Policies**, expand **Local Policies**, and then click **User Rights Assignment**. Double-click the policy **Log on as a batch job**. Click **Add User or Group** to add the user.

Tip: For details on why a check failed, select the check box against it and see the details in the description box at the bottom of the screen.

Database Information

Specify the following details of an existing Oracle Beehive database that will be used by this installation:

- [Host and Port](#)
- [Service Name](#)
- [Schema User Password](#)

Host and Port

Specify the fully qualified domain name (FQDN) of the host where the target database is running.

For example, if the host name is `foo`, its domain is `abc.com`, and the database listener port is `1521`, then enter the following:

```
foo.abc.com:1521
```

The default port on which the database listener listens for connection requests is `1521`.

To determine this port number, run the command `<Database home>/bin/lsnrctl status`. (Ensure that the environment variable `ORACLE_HOME` is set to the directory in which Oracle Database is installed and `ORACLE_SID` is set to the system identifier of your database.)

Service Name

Enter the service name for the Oracle database. This is the same as the global database name, and must be unique across all databases.

A database is identified by its *global database name*. The global database name is comprised of two parts:

```
database_name.database_domain
```

For example:

```
sales.us.yourcompany.com
```

Schema User Password

Enter the password for the Oracle Beehive database schema that is stored in the database whose information you have specified in this screen.

After clicking **Next**, the Install Wizard verifies the connection to the Oracle Database Server, initialization parameters, and other requirements. If an error occurs, then you must correct the problem before the installation process can continue.

Installation Summary

The Upgrade Summary screen presents a summary of the options you have specified so far:

- **Space Requirements:** The disk space required to install the product. The space requirement appears in a different color if the disk space available is less than what is required.

Notes: The space requirements shown on this screen do not include the space required in the target database.

The space requirements shown on this screen also might differ slightly from the actual space the component occupies on the installation disk. This is because of the differences in the disk configurations. For example, the space required on one hard drive might differ from the space required on another hard drive that uses different block size or that is managed by a different file system.

- **Inventory Location:** The location of the `oraInventory` directory where all the installation information about your product will be stored.
- **Product:** The name of the product you are installing, Oracle Beehive Administration Console.
- **Interview Details:** The details of your inputs for the other screens, such as:

- **Home Details:** The location of the Oracle home where you are installing the product.
- **Database Information:** The host name, port, and service name of the existing Oracle Beehive database that will be used by this installation process.
- **Schema Password:** The masked password for the Oracle database schema that you specified.

When you have reviewed your inputs, click **Install** to continue.

Installation Process

Use this screen to monitor the installation process. Click **Show Details** to see the details in the description box that appears

Configuration Assistants

This screen displays the configuration assistants that the Install Wizard runs to install Oracle Beehive Administration Console.

End of Installation

This screen indicates if the installation of Oracle Beehive Administration Console was successful.

Starting the Oracle Beehive Administration Console Uninstall Wizard

Start the Oracle Beehive Administration Console Uninstall Wizard by running the Oracle Beehive Administration Console Install Wizard and clicking the **Uninstall** button. Alternatively, run the Oracle Beehive Administration Console Config Wizard (or Oracle Beehive Administration Console Install Wizard) with the `-uninstall` command-line option.

Oracle Beehive Administration Console Uninstall Wizard Sequence of Screens

The following screens appear during the uninstallation of Oracle Beehive Administration Console:

- [Select Oracle Home to Uninstall](#): This screen will not appear if you started Oracle Beehive Administration Console Uninstall Wizard with Oracle Beehive Administration Console Config Wizard.
- [Start Uninstallation](#)
- [End of Uninstallation](#)

Select Oracle Home to Uninstall

This screen lists all the Oracle Beehive Administration Console instances you have installed and their respective Oracle homes.

Select the Oracle Beehive home you want to uninstall.

This screen will not appear if you started the uninstall process by running the Config Wizard. In this case, the Config Wizard will uninstall the current Oracle home.

Start Uninstallation

This screen will list actions that the Oracle Beehive Administration Console Uninstall Wizard must perform to uninstall your chosen Oracle Beehive Administration Console instance.

Note: If you started the uninstall process with the Config Wizard, the "Delete Oracle Home" action will not be displayed. This action will be performed after you have exited the Config Wizard graphical user interface.

Click **Start Uninstall** to proceed with the uninstallation of your chosen Oracle Beehive Administration Console instance.

Note: On Microsoft Windows, if you receive an error message from the Oracle Beehive Administration Console Uninstall Wizard that it was unable to delete the Oracle home folder, delete it manually, then restart your computer.

End of Uninstallation

This screen indicates that the uninstallation of the chosen Oracle Beehive Administration Console instance is successful.

Upgrading Oracle Beehive

These steps describe how to upgrade the following products:

- Oracle Beehive Release 1 (1.2.1.0.0 or 1.2.2.0.0) to Release 1 (1.3.1.0.0)
 - Oracle Beehive Provisioning Application version 1.2.1.0.0 to Oracle Beehive Provisioning Application version 1.3.1.0.0. (You may not upgrade any version of Oracle Beehive Provisioning Application earlier than 1.2.1.0.0.)
 - Oracle Beehive for DMZ version 1.2.1.0.0 or 1.2.2.0.0
1. Start the Oracle Beehive Install Wizard.
 2. In the **Select Product Type** screen, select the Oracle Beehive product you want to upgrade.
 3. After the **Select Installation Type** screen, the Oracle Beehive Installation Wizard will automatically detect if you have any existing Oracle Beehive products on the machine from which you started the Oracle Beehive Install Wizard.

If the Install Wizard detects any existing Oracle Beehive products that can be upgraded, you will see the screen **Upgrade Existing Home**. Depending on the product you want to upgrade, refer to one of the following sections:

- [Oracle Beehive Upgrade Process Sequence of Screens](#)
- [Oracle Beehive Provisioning Application Upgrade Process Sequence of Screens](#)
- [Oracle Beehive for DMZ Upgrade Process Sequence of Screens](#)

Oracle Beehive Upgrade Process Sequence of Screens

If the Oracle Beehive Wizard detects any existing Oracle Beehive Release 1 (1.2.1.0.0 or 1.2.2.0.0) instances on the machine from which you started the wizard, the following screens will appear:

- [Upgrade Existing Home](#)
- [Specify Home Location to Upgrade](#)
- [Shutdown Instances](#)
- [Database Information for Upgrade](#)
- [Upgrade Summary](#)
- [Upgrade Process](#)
- [Configuration Assistants](#)

Upgrade Existing Home

Specify if you want to upgrade the product you selected in **Select Product Type**.

If the installer detected any existing Oracle Beehive Release 1 (1.2.1.0.0 or 1.2.2.0.0) instances, specify **Yes** to upgrade one of those instances to Oracle Beehive Release 1 (1.3.1.0.0).

If you specify **No**, the Install Wizard will proceed to perform a standard installation of Oracle Beehive Release 1 (1.3.1.0.0).

Specify Home Location to Upgrade

Select from the drop down box the complete path of the Oracle Beehive instance you want to upgrade.

The drop down box will only contain Oracle Beehive home paths with the following characteristics:

- Contains an instance of Oracle Beehive Release 1 (1.2.1.0.0 or 1.2.2.0.0)
- Is registered in the central Oracle Universal Installer inventory of the machine from which you started the Install Wizard.

Shutdown Instances

Shut down all instances listed in the table. Once you have shut the instances down, click the **Rescan** button to verify that all managed instances are down. Once you have verified that all unmanaged instances (listed in red) down, select the **Manually Verified** to continue the upgrade.

Database Information for Upgrade

Specify the password of the SYS schema for the database with the specified service name.

Upgrade Summary

The Upgrade Summary screen presents a summary of the options you have specified so far:

- **Space Requirements:** The disk space required to upgrade the product. The space requirement appears in a different color if the disk space available is less than what is required.

Notes: The space requirements shown on this screen do not include the space required in the target database.

The space requirements shown on this screen also might differ slightly from the actual space the component occupies on the installation disk. This is because of the differences in the disk configurations. For example, the space required on one hard drive might differ from the space required on another hard drive that uses different block size or that is managed by a different file system.

- **Inventory Location:** The location of the `oraInventory` directory where all the installation information about your product will be stored.
- **Product:** The name of the product you are upgrading, Oracle Beehive Release 1

- **Interview Details:** The details of your inputs for the other screens, such as:
 - **Home Details:** The location of the Oracle home that you are upgrading.
 - **Database Information:** The host name, port, service name, administrator user name, and password of the existing Oracle Database instance that will be used by this upgrade process.
 - **Enterprise Definition Information:** The detailed enterprise site information that identifies this upgrade.
 - **Schema Password:** The masked password for the Oracle database schema that you specified.

When you have reviewed your inputs, click **Upgrade** to continue.

Upgrade Process

Use this screen to monitor the upgrade process. Click **Show Details** to see the details in the description box that appears.

Configuration Assistants

This screen displays the configuration assistants that the Oracle Beehive Install Wizard runs to upgrade Oracle Beehive.

The Oracle Beehive Install Wizard performs the following configuration assistants:

- **Upgrading Central Inventory:** Upgrades the product metadata to the proper version (1.3.1.0.0)
- **Applying oneoff patches:** Applies patches specific to this release
- **Upgrading Beehive Schemas:** Upgrades the Oracle Beehive schemas within the Oracle Beehive database
- **Upgrading Beehive Midtier:** Upgrades the Oracle Beehive application tier configuration

End of Upgrade

This screen indicates if the upgrade of Oracle Beehive is successful.

Oracle Beehive Provisioning Application Upgrade Process Sequence of Screens

If the Oracle Beehive Wizard detects that Oracle Beehive Provisioning Application version 1.2.1.0.0 is installed on the machine from which you started the wizard, the following screens will appear:

- [Upgrade Existing Home](#)
- [Specify Home Location to Upgrade](#)
- [Database Information for Upgrade](#)
- [Upgrade Summary](#)
- [Upgrade Process](#)
- [Configuration Assistants](#)
- [End of Upgrade](#)

Upgrade Existing Home

Specify if you want to upgrade the product you selected in **Select Product Type**.

If the installer detected that Oracle Beehive Provisioning Application version 1.2.1.0.0 is installed in your machine, specify **Yes** to upgrade it to version 1.3.1.0.0.

If you specify **No**, the Install Wizard will proceed to install Oracle Beehive Provisioning Application version 1.3.1.0.0 in a new Oracle home.

Specify Home Location to Upgrade

Select from the drop down box the complete path of the Oracle Beehive product you want to upgrade.

The drop down box will only contain Oracle Beehive home paths with the following characteristics:

- Contains Oracle Beehive Provisioning Application version 1.2.1.0.0
- Is registered in the central Oracle Universal Installer inventory of the machine from which you started the Install Wizard.

Database Information for Upgrade

Specify the password of the SYS schema for the database with the specified service name.

Upgrade Summary

The Upgrade Summary screen presents a summary of the options you have specified so far:

- **Space Requirements:** The disk space required to upgrade the product. The space requirement appears in a different color if the disk space available is less than what is required.

Notes: The space requirements shown on this screen do not include the space required in the target database.

The space requirements shown on this screen also might differ slightly from the actual space the component occupies on the installation disk. This is because of the differences in the disk configurations. For example, the space required on one hard drive might differ from the space required on another hard drive that uses different block size or that is managed by a different file system.

- **Inventory Location:** The location of the `oraInventory` directory where all the installation information about your product will be stored.
- **Product:** The name of the product you are upgrading, Oracle Beehive Provisioning Application
- **Interview Details:** The details of your inputs for the other screens, such as:
 - **Home Details:** The location of the Oracle home that you are upgrading.

When you have reviewed your inputs, click **Upgrade** to continue.

Upgrade Process

Use this screen to monitor the upgrade process. Click **Show Details** to see the details in the description box that appears.

Configuration Assistants

This screen displays the configuration assistants that the Oracle Beehive Install Wizard runs to upgrade Oracle Beehive.

End of Upgrade

This screen indicates if the upgrade of Oracle Beehive Provisioning Application is successful.

Note: After upgrading Oracle Beehive Provisioning Application to version 1.3.1.0.0, a component named **1.3.1.0.0** will be added to the Oracle Grid Control software library. Re-upload `beeshiphome.zip` to one of the shiphomes in this component as described in "[Installing Oracle Beehive with Provisioning Application](#)".

Oracle Beehive for DMZ Upgrade Process Sequence of Screens

If the Oracle Beehive Wizard detects that Oracle Beehive for DMZ version 1.2.1.0.0 or 1.2.2.0.0 is installed on the machine from which you started the wizard, the following screens will appear:

- [Upgrade Existing Home](#)
- [Specify Home Location to Upgrade](#)
- [Upgrade Summary](#)
- [Upgrade Process](#)
- [Configuration Assistants](#)
- [End of Upgrade](#)

Upgrade Existing Home

Specify if you want to upgrade the product you selected in **Select Product Type**.

If the installer detected that Oracle Beehive for DMZ version 1.2.1.0.0 or 1.2.2.0.0 is installed in your machine, specify **Yes** to upgrade it to version 1.3.1.0.0.

If you specify **No**, the Install Wizard will proceed to install Oracle Beehive for DMZ version 1.3.1.0.0 in a new Oracle home.

Specify Home Location to Upgrade

Select from the drop down box the complete path of the Oracle Beehive product you want to upgrade.

The drop down box will only contain Oracle Beehive home paths with the following characteristics:

- Contains Oracle Beehive for DMZ version 1.2.1.0.0 or 1.2.2.0.0

- Is registered in the central Oracle Universal Installer inventory of the machine from which you started the Install Wizard.

Upgrade Summary

The Upgrade Summary screen presents a summary of the options you have specified so far:

- **Space Requirements:** The disk space required to upgrade the product. The space requirement appears in a different color if the disk space available is less than what is required.

Notes: The space requirements shown on this screen do not include the space required in the target database.

The space requirements shown on this screen also might differ slightly from the actual space the component occupies on the installation disk. This is because of the differences in the disk configurations. For example, the space required on one hard drive might differ from the space required on another hard drive that uses different block size or that is managed by a different file system.

- **Inventory Location:** The location of the `oraInventory` directory where all the installation information about your product will be stored.
- **Product:** The name of the product you are upgrading, Oracle Beehive for DMZ
- **Interview Details:** The details of your inputs for the other screens, such as:
 - **Home Details:** The location of the Oracle home that you are upgrading.

When you have reviewed your inputs, click **Upgrade** to continue.

Upgrade Process

Use this screen to monitor the upgrade process. Click **Show Details** to see the details in the description box that appears.

Configuration Assistants

This screen displays the configuration assistants that the Oracle Beehive Install Wizard runs to upgrade Oracle Beehive for DMZ.

End of Upgrade

This screen indicates if the upgrade of Oracle Beehive for DMZ is successful.

Installing Oracle Beehive with Provisioning Application

This module describes how to install Oracle Beehive with Oracle Beehive Provisioning Application.

Refer to the section ["Oracle Beehive Provisioning Application Sequence of Screens"](#) in ["Oracle Beehive Install Wizard"](#) for information on how to install Oracle Beehive Provisioning Application.

This process consists of the following steps:

- [Step 1: Zip Installable Version of Oracle Beehive](#)
- [Step 2: Upload Zipped Installable Version of Oracle Beehive to Oracle Grid Control Software Library](#)
- [Step 3: Run Oracle Beehive Provisioning Application](#)

Note: If you install Oracle Beehive with Oracle Beehive Provisioning Application, you may use only test certificates, not real certificates.

To change the shiphome (an installable version of Oracle Beehive) that Oracle Beehive Provisioning Application uses to install Oracle Beehive, refer to the section ["Uploading Oracle Beehive Shiphome to Oracle Grid Control"](#).

To uninstall Oracle Beehive Provisioning Application, refer to ["Uninstalling Oracle Beehive Provisioning Application"](#).

For more information about Oracle Beehive Provisioning Application and other Oracle Beehive installation scenarios, refer to ["What You Should Know Before Installing Oracle Beehive"](#).

Note: You must specify the location of the Oracle Grid Control software library before installing Oracle Beehive Provisioning Application. Refer to ["Specifying Location of Oracle Beehive Shiphome Binaries in Grid Control"](#) in ["What You Should Know Before Installing Oracle Beehive"](#) for more information.

Follow these steps if you have not specified the software library location before installing Oracle Beehive Provisioning Application:

1. Follow the directions listed in ["Specifying Location of Oracle Beehive Shiphome Binaries in Grid Control"](#).
 2. In a browser, navigate to Oracle Grid Control. Click the **Deployments** tab. Click **Deployment Procedures** (from the Deployment Procedure Manager section). Select **Oracle Beehive Release 1 Deployment** and click **Upload**. Upload the PAR file `<Oracle Management Service home>/sysman/prov/paf/beeprovs.par`.
-
-

Step 1: Zip Installable Version of Oracle Beehive

Zip into one file, named `beeshiphome.zip`, the contents of the Oracle Beehive installation media.

Use the following commands to create a zip file from the Oracle Beehive installation media:

```
cd <Oracle Beehive installation media directory>
zip -9rv beeshiphome.zip *
```

You must use the name `beeshiphome.zip` for the zip file.

Save the file `beeshiphome.zip` on a machine or host that the Oracle Management Agent can access.

Step 2: Upload Zipped Installable Version of Oracle Beehive to Oracle Grid Control Software Library

Follow these steps to upload to the Oracle Grid Control software library the zip file you created in the previous step:

1. In a browser, navigate to the instance of Oracle Enterprise Grid Control that contains Oracle Beehive Provisioning Application.
2. Click the **Deployments** tab. Click **Provisioning** in the menu bar. You will see a list of software you may provision through Oracle Grid Control.
3. In the TreeTable, expand the **Components** entry until you see **Oracle Beehive Release 1 Shiphome**. More than one shiphome entry may be available if you have installed Oracle Beehive Provisioning Application for more than one operating system. Select the shiphome you want to upload. Click **Edit**.
4. In the **Edit Component** page, click **Upload File** in the menu bar. Select **Update from Agent Machine**. Select **Target** and specify the zip file you created in the previous step.
5. In the **Host Credentials** window, enter your user name and password. Select the **Save as Preferred Credential** check box (otherwise the upload will fail).
6. Click **Finish**.

Step 3: Run Oracle Beehive Provisioning Application

Refer to "Running Oracle Beehive Provisioning Application" in "Running Oracle Beehive Provisioning Application" for information on running Oracle Beehive Provisioning Application.

Uploading Oracle Beehive Shiphome to Oracle Grid Control

You may upload a new shiphome (an installable version of Oracle Beehive) to the central software library of Oracle Grid Control. Oracle Beehive Provisioning Application uses this shiphome to install Oracle Beehive.

To upload a new shiphome to the central library of Oracle Grid Control, follow these steps:

1. Click the **Deployments** tab. Click **Provisioning** in the menu bar. You will see a list of software you may provision through Oracle Grid Control.
2. In the TreeTable, expand the **Components** entry until you see **Oracle Beehive Release 1 Shiphome**. More than one shiphome entry may be available if you have installed Oracle Beehive Provisioning Application for more than one operating system. Select the shiphome you want to replace with a new shiphome. Click **Edit**.

Caution: Do not click **Delete**.

3. In the **Edit Component** page, select **Update from Agent Machine**. Specify the zip file that contains the new shiphome. Click **Finish**.

Uninstalling Oracle Beehive Provisioning Application

Uninstall Oracle Beehive Provisioning Application with the Oracle Beehive Universal Installer, `<Oracle home>/oui/bin/runInstaller`.

Notes: You cannot uninstall the Provisioning Application with the Oracle Beehive Uninstall Wizard.

The Oracle Beehive Uninstall Wizard will not uninstall any changes made to the Oracle Grid Control repository database.

Running Oracle Beehive Provisioning Application

Oracle Beehive Provisioning Application enables you to install Oracle Beehive from Enterprise Manager Grid Control. Oracle Beehive uses the Provisioning Advisor Framework for the purpose.

The following topics are covered:

- [Introduction to Oracle Beehive Provisioning Application](#)
- [Running Oracle Beehive Provisioning Application](#)

Introduction to Oracle Beehive Provisioning Application

Oracle Beehive Provisioning Application consists of the following components:

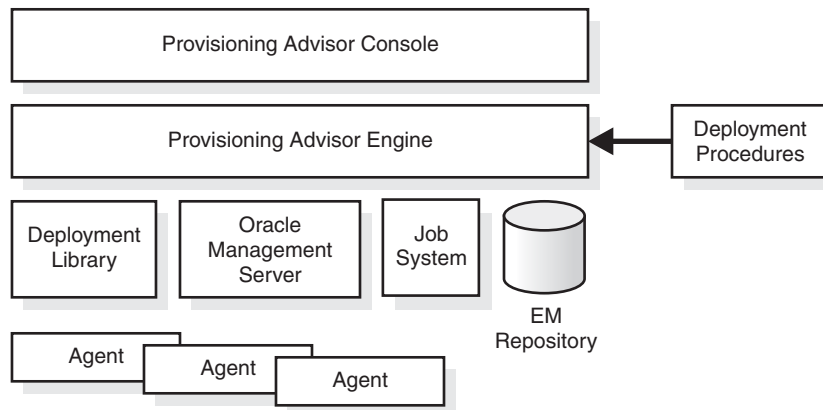
- [Provisioning Advisor Framework](#)
- [Software Library](#)
- [Job System](#)

Provisioning Advisor Framework

Provisioning Advisor Framework (PAF) provides a seamless process to integrate tasks that are associated with an application. The framework automates, orchestrates, and tracks the tasks of various applications, such as installers, upgrade assistants, and custom scripts.

PAF is built on top of the Enterprise Manager Grid Control structure. This structure comprises of Oracle Management Server (OMS), Job System, Enterprise Manager (EM) Repository, Agent and the Software Library. PAF uses an XML file called the Deployment Procedure (DP) as an input from an application. The different tasks of the application are broken down and listed in the Deployment Procedure. Each individual task is referred to as a step. A step could be further classified based on the task it performs. The step could be a manual step that requires user interaction or could be a host command step that runs a command on a specified list of targets. A group of step is referred to as a Phase. This feature enables the execution of a group of steps either in parallel or serially on the target nodes. A Deployment Procedure Engine maintains the state of the procedure while executing the phases and steps in the supplied DP.

The following figure illustrates the Provisioning Advisor Framework architecture:

Figure 8–1 Provisioning Advisor Framework Architecture**Software Library**

The Software Library is an infrastructure entity for storing and retrieving files and packages. This provides a common repository for storing software binaries, scripts and other files that are used by provisioning, cloning, and other applications.

Job System

Enterprise Manager Job System is a repository-based system that enables you to schedule and execute jobs on target machines. The Job System supports preconfigured job types, such as OS Commands and File Transfer. The Job System also enables the applications to create job types.

Running Oracle Beehive Provisioning Application

Note: If you are installing Oracle Beehive with Oracle Beehive Provisioning Application, and you are not on the sudoers list of the target machine, follow these steps:

1. Select the **Deployments** tab. Click **Deployment Procedures** (from the Deployment Procedure Manager section). Select **Oracle Beehive Release 1 Deployment** and click **Create Like**.
 2. In the **Create Like Procedure** page, fill in the fields as follows:
In the TreeTable, select **Verify and fix Beehive system requirements**. Change the **Run Privilege** to **Normal** (from sudo).
 3. Click **Save**.
 4. Follow the steps in this section, except run this copy of Oracle Beehive Provisioning Application.
-

Follow these steps to run Oracle Beehive Provisioning Application:

1. Select the **Deployments** tab.
2. Click **Deployment Procedures** (from the Deployment Procedure Manager section).
3. Select **Oracle Beehive Release 1 Deployment**.
4. Click **Schedule Deployment...**

Oracle Beehive Provisioning Application will present you with the following pages in which you will enter required parameters for Oracle Beehive installation:

1. [Target List Page](#)
2. [Install Inputs Page](#)
3. [Credentials/Schedule Page](#)
4. [Summary Page](#)

Note: The parameters are similar to the installation procedure described in "[Oracle Beehive \(Standard Installation\) Sequence of Screens](#)" in "[Oracle Beehive Install Wizard](#)" in the Oracle Beehive Installation Guide of your operating system.

Target List Page

The following describes the parameters for the **Target List** page:

- **Host targets for Oracle Beehive installation:** Select the hosts on which you want to install Oracle Beehive. You may select more than one host, add a host to the list, and remove a host from the list.

Install Inputs Page

The following describes the parameters for the **Install Inputs** page:

- Install Input
 - **Oracle home:** Specify the directory where you want to install Oracle Beehive. You must specify a new Oracle home directory for each new installation of Oracle Beehive.
- Database Details: Specify the following details of an existing Oracle Database instance that will be used by this installation. Refer to the module "[Oracle Beehive Database Requirements](#)" in the Oracle Beehive Installation Guide of your operating system for database requirements for Oracle Beehive Release 1.

Oracle RAC Notes: If you are specifying details for an Oracle Real Application Cluster (Oracle RAC) database, ensure that you have entered the following information correctly; Oracle Beehive Provisioning Application will not verify it by trying to connect to each database host. Instead, Oracle Beehive Provisioning Application constructs the connect descriptor with the database information you provide, then verifies if the connect descriptor can reach the database.

In addition, for Oracle Beehive Release 1 (1.2) and earlier, if you are specifying an Oracle RAC database, you must enable SSL Oracle Notification Services (ONS) notification. For more information, refer to ["Enable Secure ONS Notification"](#) in ["Configuring and Installing Oracle Beehive for Oracle RAC"](#). (For Oracle Beehive Release 1 (1.3) and later, this is optional.)

If you are installing Oracle Beehive in a high availability environment (you are installing multiple instances of Oracle Beehive) with the provisioning application, and you are using an Oracle RAC database, you may only use test certificates; you may not use real certificates in this situation.

- **Hostname and Port:** Specify the fully qualified domain name (FQDN) of the host where the target database is running followed by the database listener port.

For example, if the host name is `foo`, its domain is `abc.com`, and the database listener port is `1521`, then you must enter the following:

```
foo.abc.com:1521
```

Note: In case of an Oracle RAC database, use the following format:

```
virtual_host1:virtual_port1^virtual_host2:virtual_port2^ ...
```

- **Service Name:** Enter the service name for the Oracle Database. This is the same as the global database name, and must be unique across all databases. This is also the same as the Service Name the Oracle Beehive Install Wizard asks you when you choose to install Oracle Beehive against an existing database.

In the Oracle Beehive Installation Guide of your operating system, refer to ["Installing Oracle Beehive Against an Existing Database"](#) for more information. Refer to ["Database Information"](#) in ["Oracle Beehive Install Wizard"](#) for more information about the service name.

- **Username:** Specify the administrative user name (that can perform database administration tasks) for the database instance. Typically, this user is the `SYS` account.
- **Password:** Enter the password for the Oracle database administrative user account that you specified in the **Username** field.
- Enterprise Information
 - **Enterprise:** Specify the name for the new enterprise you want to create. An enterprise is the top-level container for all users and objects. An enterprise may contain any number of organizations and workspaces.

The enterprise name is typically a company name, such as `Oracle`.

In Oracle Beehive Release 1, an Oracle Beehive deployment may have only one enterprise.

- **Organization:** Specify the name of the organization to be created and added to the enterprise you specified previously. An organization is a logical grouping of users, groups, workspaces, and resources at a level lower than the enterprise level. Later, you may add any number of organizations to the enterprise you specified previously.

The organization is typically a sub-level or department under the enterprise, such as `Sales` or `Accounting`.

- **Instance Name:** Specify the name of the instance for this Oracle Beehive installation. The instance name is used during server configuration.

The full name of an Oracle Beehive instance consists of the instance, host, and domain names. For example, if you name your instance `myinstance1`, your host is `myhost`, and your domain is `example.com`, the full name of your instance will be `myinstance1.myhost.example.com`.

Note: Only enter the instance name; do not specify the host and domain name. Oracle Beehive will append the host and domain name to the instance name. For example, you would specify `myinstance1`, not `myinstance1.myhost.example.com`.

All Oracle Beehive Release 1 instances must have a unique name.

- **Site:** Specify the name of the site. A site consists of all Oracle Beehive services and service instances and the database they use.

A site is typically a reference to a physical location, such as `Headquarters` or `Montreal`.

In Oracle Beehive Release 1, an Oracle Beehive deployment may have only one site.

- **beeadmin Credentials**
 - **Password:** Specify and confirm the password for the `beeadmin` user. The `oc4jadmin` and `bpeladmin` users will have the same password.
 - **Confirm Password:** Re-enter the password you entered in the previous field.
- **Oracle Beehive Schema Information**
 - **Password:** Use this screen to specify and confirm the password for the new database schema that will be created and used by the Oracle Beehive instance.
 - **Confirm Password:** Re-enter the password you entered in the previous field.

Credentials/Schedule Page

The following describes the parameters for the **Credentials/Schedule** page:

- **Oracle Home Credentials:** The user credentials you will specify on this page must already exist on the target nodes. Also, ensure that the specified user is a part of the `osinstall` group. In addition, this user requires `sudo` privileges to run certain scripts as the `root` user on these target nodes.

Note: Ensure that you have entered the following user credentials correctly; Oracle Beehive Provisioning Application will not verify them.

- **Specify Credentials for:** Specify the Oracle homes that the Provisioning Application will access with the credentials (username and password) that you will specify in the following fields.
 - * **Username, Password, and Confirm Password:** Specify the username and the password of the user that can access the Oracle homes you specified in **Specify Credentials for**.
- Schedule
 - **One Time (Immediately):** Starts the installation immediately
 - **One Time (Later):** Starts the installation at the specified time. If you do not want the procedure to start immediately, select this option and use the **Time Zone, Start Date, and Start Time** to schedule the procedure. If you change the date or time to a time in the future, the procedure will start on the specified date and time.
 - Grace Period
 - * **Indefinite:** Use this option if you want the Oracle Management Service to continue to attempt to start the patching job for an infinite amount of time.
 - * **End After *n* Hours *m* Minutes:** In addition, you can specify the **End After** option of **Grace Period** to allow the Oracle Management Service to continue to attempt to start the job up to the specified time. This is useful, if the Oracle Management Agent is down and the Oracle Management Service is unable to start the patching job, because if the patching job is not started within the allotted time frame, the Oracle Management Service marks the patching job as skipped.

Summary Page

Review the installation parameters in the **Summary** page and click **Finish** to start the installation of Oracle Beehive.

Installing Oracle Beehive in Silent Mode (Non-Interactive)

This module describes how to install Oracle Beehive in silent mode, which allows minimal input from the user. This process involves creating a *response file*, which is an XML file that contains values required by the Oracle Beehive Install Wizard.

Use this process if you want to install Oracle Beehive with a batch process or do not want to use the GUI provided by the Oracle Beehive Install Wizard.

This module covers the following silent mode processes:

- [Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard](#)
- [Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard and Config Wizard](#)
- [Installing Oracle Beehive Provisioning Application in Silent Mode](#)
- [Installing Oracle Beehive for DMZ in Silent Mode](#)
- [Verifying Oracle Beehive Installation](#)
- [Recovering from Failed Oracle Beehive Configuration](#)
- [Upgrading Oracle Beehive in Silent Mode](#)
- [Uninstalling Oracle Beehive in Silent Mode](#)

Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard

This process installs and configures Oracle Beehive by running the Oracle Beehive Install Wizard only once. It consists of the following steps:

1. [Verify that Oracle Database is installed and running](#)
2. [Create a response file](#)
3. [Start the installation and configuration in silent mode](#)
4. [Verify the installation](#)

Step 1 Verify that Oracle Database is installed and running

To verify that Oracle Database is running, connect to it and run a query as follows:

1. Ensure that the environment variables `ORACLE_HOME` is set to the home directory of Oracle Database and `ORACLE_SID` is set to the SID of the database you want to configure with this installation of Oracle Beehive.

2. Run the following commands to check the version of Oracle Database and verify that it is running (<Database home> is where you installed Oracle Database):

```
<Database home>/bin/sqlplus '/ as sysdba'
```

```
SQL> select version from v$instance;
```

```
VERSION
```

```
-----
```

```
10.2.0.3.0
```

```
SQL> exit;
```

3. Verify that the listener is listening to a service with the same name as ORACLE_SID:

```
<Database home>/bin/lsnrctl status
```

Note: Refer to "[Oracle Beehive Database Requirements](#)" for more information about minimum database requirements.

Step 2 Create a response file

A response file is an XML file that contain values required for installation and configuration.

To create a response file, edit the response file according to the comments in it. Use one of the following methods to create a response file:

- Use the existing response file, `response/bee hive_response_template.xml` in the installation media.
- Generate this response file in your home directory with the following command (from the installation media):

```
runInstaller -generateResponseTemplate
```

Tip: Create a fully configured response file by running the Oracle Beehive Install Wizard (not in silent mode) until you reach the Installation Summary screen. Click **Save**. The Oracle Beehive Install Wizard saves the inputs you entered and options you selected in a response file.

Note that the Oracle Beehive Install Wizard **will not save** the passwords you entered in the response file. You must edit this response file and add the required passwords.

Step 3 Start the installation and configuration in silent mode

Run the following command (from the installation media) to start the installation and configuration of Oracle Beehive in silent mode:

```
runInstaller -responseFile full_path_of_response_file -silent
```

After the installation and configuration are completed, Oracle Beehive servlet URLs are displayed. You will also find this information in a log file with a name similar to <Oracle inventory>/logs/installActions*.log (where <Oracle inventory> is the Oracle inventory directory).

If there is no Oracle inventory directory (the directory that stores information about the Oracle products in your computer) or this is the first installation of an Oracle

product in the computer, you will be asked to run the `<Oracle inventory>/oraInstRoot.sh` script as the root user.

Step 4 Verify the installation

Run the administration tool `beectl` as described in ["Verifying Oracle Beehive Installation"](#).

Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard and Config Wizard

This process consists of two phases: an installation-only (or software-only) phase that installs Oracle Beehive by running the Oracle Beehive Install Wizard and a configuration-only phase that configures Oracle Beehive by running the Oracle Beehive Config Wizard.

This process consists of the following steps:

1. [Verify that Oracle Database installed and running](#)
2. [Create a response file](#)
3. [Start the installation-only phase in silent mode](#)
4. [Start the configuration-only phase in silent mode](#)
5. [Verify the installation](#)

Step 1 Verify that Oracle Database installed and running

This step is the same as [Step 1, Verify that Oracle Database is installed and running in "Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard"](#).

Step 2 Create a response file

This step is the same as [Step 2, Create a response file in "Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard"](#), except you only need to specify the values `homeLocation` and `softwareOnly`. (For the configuration-only phase, Step 4, you will need to specify all the other values.) The following is an excerpt from a response file that has these two values specified:

```
*****
Oracle Home Information
*****
-->
<object class="oracle.ocs.install.wizard.framework.beans.HomeInfoBean">
  <!--
  # This string property holds the oracle home location.
  # Specify an absolute path here.
  # The path should not contain special characters
  -->
  <void property="homeLocation">
    <string>/my_oracle_home</string>
  </void>
</object>

<!--
*****
Install Type Information
*****
-->
```

```

<object class=
  "oracle.ocs.install.wizard.product.main.domain.InstallTypeInfoBean">
  <!--
  # This boolean property holds the type of install.
  # Specify true to install just the software only or
  # specify false to install and configure.
  # Defaults to false if not specified.
  -->
  <void property="softwareOnly">
    <boolean>true</boolean>
  </void>
</object>

```

Step 3 Start the installation-only phase in silent mode

Run the following commands (from the installation media) to start the installation-only phase in silent mode:

```
runInstaller -responseFile full_path_of_response_file -silent
```

Step 4 Start the configuration-only phase in silent mode

Wait until the installation-only phase is finished and successful. Ensure that you have specified all required values in the response file. Run the following commands to start the configuration-only phase (<Oracle home> is where you installed Oracle Beehive):

```
cd <Oracle home>/beehive/oobwiz
./configWizard -responseFile full_path_of_response_file -silent
```

Log files with names similar to <Oracle home>/beehive/oobwiz/logs/configActions*.log contain information about this configuration-only phase.

If there is no Oracle inventory directory (the directory that stores information about the Oracle products in your computer) or this is the first installation of an Oracle product in the computer, you will be asked to run the <Oracle inventory>/oraInstRoot.sh script as the root user.

Step 5 Verify the installation

Run the administration tool `beectl` as described in "[Verifying Oracle Beehive Installation](#)".

Installing Oracle Beehive Provisioning Application in Silent Mode

You may install Oracle Beehive Provisioning Application in silent mode.

The process is similar to "[Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard](#)" except that you do not verify that a database is running and you use the response file (from the installation media) `response/provplugin_response_template.xml`. You can also generate this response file with the command `runInstaller -generateResponseTemplate`. The response file will be in your home directory.

Installing Oracle Beehive for DMZ in Silent Mode

You may install Oracle Beehive for DMZ in silent mode.

The process is similar to ["Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard"](#) except that you use the response file (from the installation media) `response/dmz_response_template.xml`. You can also generate this response file with the command (from the installation media) `runInstaller -generateResponseTemplate`. The response file will be in your home directory.

Verifying Oracle Beehive Installation

Run the `beectl status` command:

```
<Oracle home>/beehive/bin/beectl status
```

If Oracle Beehive is installed and configured successfully, you should see output similar to the following:

Component identifier	Component type	Status
BTI_instance1.example.com	Bti	RUNNING
BEEEMGMT_instance1.example.com	ManagedOc4j	RUNNING
BEEAPP_instance1.example.com	ManagedOc4j	RUNNING
oc4j_soa_instance1.example.com	ManagedOc4j	RUNNING
BEECORE_instance1.example.com	ManagedOc4j	RUNNING
ohs_instance1.example.com	HttpServer	RUNNING

If some of these components have not been started, review the log files in the following directories:

- `<Oracle inventory>/logs`
- `<Oracle home>/beehive/logs`

Recovering from Failed Oracle Beehive Configuration

If Oracle Beehive configuration failed, run the `configWizard` command. This command will automatically detect that Oracle Beehive configuration was not successful. It will then run only the configuration tools that failed or were aborted.

Upgrading Oracle Beehive in Silent Mode

You may upgrade the following Oracle Beehive products in silent mode:

- Oracle Beehive Release 1 (1.2.1.0.0 or 1.2.2.0.0) to Release 1 (1.3.1.0.0)
- Oracle Beehive Provisioning Application version 1.2.1.0.0 to Oracle Beehive Provisioning Application version 1.3.1.0.0. (You may not upgrade any version of Oracle Beehive Provisioning Application earlier than 1.2.1.0.0.)
- Oracle Beehive for DMZ version 1.2 to version 1.3.1.0.0
- Oracle Collaboration Coexistence Gateway version 1.2 to version 1.3.1.0.0
- Oracle Beehive Release 1 (1.2.1.0.0 or 1.2.2.0.0) to Release 1 (1.3.1.0.0)

- Oracle Beehive Provisioning Application version 1.2.1.0.0 to Oracle Beehive Provisioning Application version 1.3.1.0.0. (You may not upgrade any version of Oracle Beehive Provisioning Application earlier than 1.2.1.0.0.)
- Oracle Beehive for DMZ version 1.2 to version 1.3.1.0.0
- Oracle Collaboration Coexistence Gateway version 1.2 to version 1.3.1.0.0

The upgrade process to upgrade any of these Oracle Beehive products is the same as "[Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard](#)" except that you create a response file from a different template depending on the product you are upgrading. The following lists the names of the response file templates to use for each Oracle Beehive product to upgrade:

- Oracle Beehive Release 1 (1.2.1.0.0 or 1.2.2.0.0): `beehive_upgrade_response_template.xml`
- Oracle Beehive Provisioning Application version 1.2.1.0.0: `provplugin_upgrade_response_template.xml`
- Oracle Beehive for DMZ version 1.2: `dmz_upgrade_response_template.xml`
- Oracle Collaboration Coexistence Gateway version 1.2: `exconnector_upgrade_response_template.xml`

Uninstalling Oracle Beehive in Silent Mode

You may uninstall Oracle Beehive in silent mode.

The process is similar to "[Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard](#)" except for the following:

- You do not need to verify that a database is running.
- Use the response file (from the installation media) `/response/beehive_uninstall_response_template.xml`. You can also generate this response file with the command `runInstaller -generateResponseTemplate`. The response file will be in your home directory.
- Run the following command (from the installation media):

```
runInstaller -uninstall  
-responseFile full_path_of_uninstall_response_file -silent
```

You may also run the following command:

```
<Oracle home>/beehive/oobwiz/configWizard -uninstall  
-responseFile full_path_of_uninstall_response_file -silent
```

Oracle Beehive Install and Config Wizard Command-Line Options

This module describes command-line options for the Oracle Beehive Install Wizard and Config Wizard.

Oracle Beehive Install Wizard

Starts the installation wizard, which in turn can start the configuration wizard.

Usage

```
runInstaller [-option name]
```

Options

The following table describes the Oracle Beehive Install Wizard command-line options.

Table 10–1 *runInstaller Options*

Option	Description
-debug	Turns on all levels of logging.
-entryPoint <i>entry_point</i>	Specifies the product-specific prerequisite set that will be run. For Oracle Beehive, this is oracle.bee_allProducts.
-force	Allows silent mode installation into a non-empty directory. Valid for install time use only.
-generateResponseTemplate	Generates response files (for Oracle Beehive installation, Oracle Beehive Provisioning Application, and Oracle Beehive uninstallation) in the user's home directory.
-help	Outputs help about runInstaller command-line arguments.

Table 10–1 (Cont.) runInstaller Options

Option	Description
<code>-invPtrLoc <i>absolute_path_to_oraInst.loc_file</i></code>	<p>UNIX-based systems only: Specifies the location of an <code>oraInst.loc</code> file other than the default (which is <code>/etc/oraInst.loc</code>).</p> <p>For Solaris operating system (SPARC 64-Bit): Specifies the location of an <code>oraInst.loc</code> file other than the default (which is <code>/var/opt/oracle/oraInst.loc</code>).</p> <p>The <code>oraInst.loc</code> file contains the following lines:</p> <pre>inventory_loc=inventory_directory_path inst_group=INVGROUP</pre> <p><code>INVGROUP</code> is the UNIX OS group that has write access to the inventory directory.</p>
<code>-logLevel <i>level</i></code>	Filters messages that have a lesser priority level than <i>level</i> . Valid levels are severe, warning, info, config, fine, finer, and finest.
<code>-noconsole</code>	Windows only: Suppresses display of messages to the console.
<code>-nowait</code>	Windows only: Does not prompt the user for exit confirmation.
<code>-prereq_config_loc <i>location</i></code>	Specifies the fully qualified path to the prereq directory that contains the prerequisite.properties file.
<code>-prereqcheck</code>	Runs the prerequisite checker only.
<code>-record</code>	Records the user's response from the installation wizard to a response file. Specify the name of the file to be created with the <code>-responseFile</code> option.
<code>-responseFile <i>path</i></code>	Specifies the response file and path to use.
<code>-silent</code>	Installs Oracle Beehive in silent mode operation. You must also specify a response file with the <code>-responseFile</code> option.
<code>-target_loc <i>directory</i></code>	Checks available disk space in <i>directory</i> .
<code>-uninstall</code>	<p>Launches the wizard in uninstall mode only if any product homes are detected in the install inventory.</p> <p>Silent uninstallation can be performed by additionally using the options <code>-silent</code> and <code>-responseFile <i>uninstall_responsefile</i></code>.</p> <p>Windows only: This option will work only from <code>setup.exe</code> on the installation media.</p>
<code>-waitforcompletion</code>	Causes the installer process to execute in the foreground; it will not return until execution completes.

Oracle Beehive Config Wizard

Starts the configuration wizard. The full path of the command is `<Oracle home>/beehive/oobwiz/configWizard`.

Usage

```
configWizard [-option name]
```

Options

The following table describes the configWizard command options:

Table 10–2 configWizard Options

Option	Description
-entryPoint <i>entry_point</i>	Specifies the product-specific prerequisite set that will be run. For Oracle Beehive, this is oracle.bee_allProducts.
-force	Allows silent mode installation into a non-empty directory. Valid for install time use only.
-generateResponseTemplate	Generates response files in user's home directory.
-help	Outputs help about configWizard command-line arguments.
-invPrtLoc <i>absolute_path_to_oraInst.loc_file</i>	UNIX-based systems only: Specifies the location of an oraInst.loc file other than the default (which is /etc/oraInst.loc). For Solaris operating system (SPARC 64-Bit): Specifies the location of an oraInst.loc file other than the default (which is /var/opt/oracle/oraInst.loc). The oraInst.loc file contains the following lines: inventory_loc= <i>inventory_directory_path</i> inst_group= <i>INVGROUP</i> <i>INVGROUP</i> is the UNIX OS group that has write access to the inventory directory.
-logLevel <i>level</i>	Filters log messages that have a lesser priority level than <i>level</i> . Valid levels are: severe, warning, info, config, fine, finer, finest.
-prereq_config_loc <i>location</i>	Specifies the fully qualified path to the prereq directory that contains the prerequisite.properties file.
-prereqcheck	Runs only the prerequisite checker.
-record	Records the user's response from the installation wizard to a response file. Specify the name of the file to be created with the -responseFile option.
-reponseFile <i>path</i>	Specifies the response file and path to use.
-silent	Configures Oracle Beehive in silent mode operation. You must also specify a response file with the -responseFile option.
-target_loc <i>directory</i>	Checks available disk space in <i>directory</i> .
-uninstall	Launches the wizard in uninstall mode only if any product homes are detected in the install inventory. Silent uninstallation can be performed by additionally using the options -silent and -responseFile <i>uninstall_responsefile</i> . Windows only: You cannot perform uninstallations with the configWizard. This includes silent uninstallations. To uninstall Oracle Beehive, run setup.exe -uninstall from the installation media.

Oracle Beehive Uninstall Wizard

You may uninstall the following products with the Oracle Beehive Uninstall Wizard:

- Oracle Beehive Release 1
- Oracle Beehive Provisioning Application
- Oracle Beehive for DMZ
- Oracle Beehive Integration for Zimbra

The following topics about uninstalling Oracle Beehive are covered:

- [Starting the Oracle Beehive Install Wizard](#)
- [Oracle Beehive Uninstall Wizard Sequence of Screens](#)
- [Uninstalling Multiple Instances](#)
- [Manually Uninstalling Oracle Beehive](#)
- [Manually Deleting Oracle Beehive Tablespaces and Datafiles](#)

Starting the Oracle Beehive Install Wizard

Start the Oracle Beehive Uninstall Wizard by running the Oracle Beehive Install Wizard and clicking the **Uninstall** button. Alternatively, run the Oracle Beehive Config Wizard (or Oracle Beehive Install Wizard) with the `-uninstall` command-line option.

Oracle Beehive Uninstall Wizard Sequence of Screens

The following screens appear during the uninstallation of Oracle Beehive:

- [Select Oracle Home to Uninstall](#): This screen will not appear if you start Oracle Beehive Uninstall Wizard with Oracle Beehive Config Wizard.
- [Option to Remove Database Schema](#): This screen will only appear if you are uninstalling Oracle Beehive Release 1 and it is the last one configured in your database.
- [Start Uninstallation](#)
- [End of Uninstallation](#)

Select Oracle Home to Uninstall

This screen lists all the Oracle Beehive products you have installed and their respective Oracle homes.

Select the Oracle Beehive home you want to uninstall.

This screen will not appear if you started the uninstall process by running the Config Wizard. In this case, the Config Wizard will uninstall the current Oracle Beehive home.

Option to Remove Database Schema

You will see this screen if you are uninstalling Oracle Beehive Release 1 and it is the last instance configured in your database.

Choose from one of the following options:

Remove Oracle Beehive Instance Only

This option will only deconfigure the Oracle Beehive instance.

Remove Oracle Beehive and Schema

This option will deconfigure the Oracle Beehive instance and remove the Oracle Beehive schema from the database. If you choose this option, you must provide the following information:

- **Admin User Name:** Specify the administrative user name (that can perform database administration tasks) for the database instance. Typically, this user is the SYS account.
- **Admin User Password:** Enter the password for the Oracle database administrative user account that you specified in the **Admin User Name** field.

Start Uninstallation

This screen will list actions that the Oracle Beehive Uninstall Wizard must perform to uninstall your chosen Oracle Beehive product.

Note: If you started the uninstall process with the Config Wizard, the "Delete Oracle Home" action will not be displayed. This action will be performed after you have exited the Config Wizard graphical user interface.

Click **Start Uninstall** to proceed with the uninstallation of your chosen Oracle Beehive product.

Note: On Microsoft Windows, if you receive an error message from the Oracle Beehive Uninstall Wizard that it was unable to delete the Oracle home folder, delete it manually, then restart your computer.

End of Uninstallation

This screen indicates that the uninstallation of the chosen Oracle Beehive product is successful.

Uninstalling Multiple Instances

If you wish to uninstall multiple instances of Oracle Beehive (which share the same database), uninstall each instance one at a time; do not run the Uninstall Wizard of each instance at the same time.

If you have multiple instances that share the same database, and you run the Uninstall Wizard of each instance at the same time, each Uninstall Wizard will assume that the instance it is uninstalling is not the last one to be uninstalled. As a result, the Oracle Beehive schema will not be removed from the database.

Manually Uninstalling Oracle Beehive

If you are unable to uninstall Oracle Beehive with the Uninstall Wizard because it was not able to remove the Oracle Beehive schemas from the database, follow these steps:

1. Stop all Oracle Beehive processes:

```
<Oracle Beehive home>/beehive/bin/beectl stop -all
```

2. Manually stop any Oracle Beehive processes:

Linux and UNIX-Based Operating Systems:

- a. Look up remaining Oracle Beehive processes:

```
ps -ef --forest --columns 2000 | grep beehive
```

- b. Kill any remaining Oracle Beehive processes.

3. Deinstall the Oracle Beehive repository with the following command:

```
perl schema_install.pl
  -deinstall
  -oracle_home <database home>
  -connect_string '<database connect string>'
```

The Perl script `schema_install.pl` is located in `<Oracle Beehive home>/beehive/db`. Perl can be found in `<Oracle Beehive home>/perl`.

4. Uninstall Oracle Beehive with Oracle Universal Installer. Run `<Oracle Beehive home>/oui/bin/runInstaller`.

Manually Deleting Oracle Beehive Tablespaces and Datafiles

To manually delete Oracle Beehive tablespaces and datafiles, run the following SQL*Plus commands as a user with SYSDBA privileges. Ensure that the environment variable `ORACLE_SID` is set to the SID of the database that contains the Oracle Beehive tablespaces and datafiles.

Note: You may need to manually delete Oracle Beehive database users. Run the SQL command `DROP USER <user name> CASCADE` for the following users:

- BEE_DATA
 - BEE_CDCPUB
 - BEE_CODE
 - ORABPEL
 - ORAWSM
 - ORAESB
-
-

1. SQL> **SET LINE 1000 PAGES 0**

This command sets the line size to a large number and suppresses other formatting information so that the next command can output additional SQL*Plus commands without any line breaks.

```
2. SQL> SELECT 'DROP TABLESPACE ' || tablespace_name || '
INCLUDING CONTENTS AND DATAFILES;' FROM dba_tablespaces WHERE
tablespace_name LIKE 'BEE%';
```

This command outputs a list of tablespace DROP commands. Review each command before running them.

The following is an example of running these two commands, then running each of the generated tablespace DROP commands:

```
SQL> SET LINE 1000 PAGES 0
SQL> SELECT 'DROP TABLESPACE ' || tablespace_name || ' INCLUDING CONTENTS AND
DATAFILES;' FROM dba_tablespaces WHERE tablespace_name LIKE 'BEE%';
DROP TABLESPACE BEE_ARCHIVE INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE BEE_AUDIT INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE BEE_DATA INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE BEE_INDEX INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE BEE_INTERFACE INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE BEE_LOBS INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE BEE_QUEUES INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE BEE_SEARCH_DATA INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE BEE_SEARCH_LOBS INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE BEE_SEED INCLUDING CONTENTS AND DATAFILES;
```

11 rows selected.

```
SQL> DROP TABLESPACE BEE_ARCHIVE INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_AUDIT INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_DATA INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_INDEX INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_INTERFACE INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_LOBS INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_QUEUES INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_SEARCH_DATA INCLUDING CONTENTS AND DATAFILES;
```


Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_SEARCH_INDEX INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_SEARCH_LOBS INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

```
SQL> DROP TABLESPACE BEE_SEED INCLUDING CONTENTS AND DATAFILES;
```

Tablespace dropped.

Manually Deleting Customized Oracle Beehive Customized Tablespaces and Datafiles

If you have customized the Oracle Beehive tablespace layouts, run the following SQL*Plus commands as a user with SYSDBA privileges:

1. SQL> **SET LINE 1000 PAGES 0**

This command sets the line size to a large number and suppresses other formatting information so that the next command can output additional SQL*Plus commands without any line breaks.

2. SQL> **SELECT 'DROP TABLESPACE "' || ts_name || "' INCLUDING CONTENTS AND DATAFILES;' AS "Drop tablespace commands" FROM sys.bee_tablespaces;**

This command generates a list of tablespace DROP commands. Review each command before running them.

3. SQL> **DROP VIEW sys.bee_tablespaces;**

This command drops the view that contains the customized tablespaces.

Part II

Oracle Beehive Post-Installation Configuration

This part describes how to configure Oracle Beehive. In particular, it describes how to configure security-related issues, install Oracle Beehive Integration for Outlook, and create Oracle Beehive clones ideal for testing changes to an Oracle Beehive instance before applying them to a production instance.

This part contains the following chapters:

- [Oracle Beehive Post-Installation Procedures](#)
- [Configuring Oracle Beehive Demilitarized Zone Instances](#)
- [Integrating and Synchronizing LDAP with Oracle Beehive](#)
- [Configuring SSL](#)
- [Configuring TLS with Oracle Wallet](#)
- [Configuring SSL for LDAP Integration](#)
- [Enabling AJPS](#)
- [Configuring E-Mail with SSL](#)
- [Configuring Oracle Secure Enterprise Search](#)
- [Configuring Single Sign-On with Oracle Beehive](#)
- [Configuring External Oracle BPEL Process Manager with Oracle Beehive](#)
- [Installing Oracle Beehive Integration for Outlook](#)
- [Cloning Oracle Beehive Application Tiers and Sites](#)
- [Oracle Beehive Administration Console Post-Installation Procedures](#)

Oracle Beehive Post-Installation Procedures

Depending on the installation scenario you have chosen, your security requirements, the standards-based clients you want to use with Oracle Beehive, or any other issue particular to your deployment, perform one or more of the following procedures:

- [Using Oracle Beehive Command-Line Utility](#)
- [Configuring Oracle Beehive to Listen on Ports Less Than 1024](#)
- [Opening Ports Required by Oracle Beehive for FTP](#)
- [Configuring DMZ Instances](#)
- [Integrating and Synchronizing LDAP with Oracle Beehive](#)
- [Configuring SSL](#)
- [Configuring TLS](#)
- [Configuring SSL for LDAP Integration](#)
- [Enabling AJP](#)
- [Configuring Oracle Beehive E-mail](#)
- [Configuring Oracle Secure Enterprise Search](#)
- [Configuring Oracle Single Sign-On](#)
- [Configuring External Oracle BPEL Process Manager with Oracle Beehive](#)
- [Installing Oracle Beehive Integration for Outlook](#)
- [Configuring Oracle Beehive Integration for Zimbra](#)
- [Cloning Oracle Beehive](#)
- [Performing Oracle Beehive Administration Console Post-Installation Procedures](#)

Using Oracle Beehive Command-Line Utility

Many of the following post-installation procedures use the Oracle Beehive command-line utility `beectl`. For more information about `beectl`, refer to "Oracle Beehive Command-Line Utility" in *Oracle Beehive Administrator's Reference Guide*.

Configuring Oracle Beehive to Listen on Ports Less Than 1024

Follow one of these series of steps to change a port number to which Oracle Beehive listens to a privileged port number (less than 1024):

- [Changing HTTP Port](#)

- [Changing Other Ports](#)
- [Enabling BPEL Workflows to Use New Port Number](#)

Changing HTTP Port

Follow these steps to change the HTTP port to 80 or HTTPS port to 443 (or any other port less than 1024):

1. As the root user, change the owner of `.apachectl` to the root user, set its user ID to `root`, and give all users permission to execute it::

```
chown root <Oracle home>/Apache/Apache/bin/.apachectl
chmod a+x <Oracle home>/Apache/Apache/bin/.apachectl
chmod a+s <Oracle home>/Apache/Apache/bin/.apachectl
```

Tip: Alternatively, instead of changing to the root user, you may use the `sudo` command instead:

```
sudo chown root <Oracle home>/Apache/Apache/bin/.apachectl
sudo chmod a+x <Oracle home>/Apache/Apache/bin/.apachectl
sudo chmod a+s <Oracle home>/Apache/Apache/bin/.apachectl
```

2. Determine the name of the listening component and property name of the port you want to change. Run the `beectl list_ports` command to list all available ports (the `--format` option is optional):

```
beectl list_ports --format xml
```

```
...
<row>
  <column name="Protocol">HTTP</column>
  <column name="Listening Port">7777</column>
  <column name="Virtual Port">7777</column>
  <column name="Defining Component">ohs_site1.example.com</column>
  <column name="Property Name">HttpListenPort</column>
  <column name="Listening Component">ohs_site1.example.com</column>
</row>
<row>
  <column name="Protocol">HTTPS</column>
  <column name="Listening Port">4443</column>
  <column name="Virtual Port">4443</column>
  <column name="Defining Component">ohs_site1.example.com</column>
  <column name="Property Name">HttpSslListenPort</column>
  <column name="Listening Component">ohs_site1.example.com</column>
</row>
```

In the previous example, the HTTP (and HTTPS) listening component is `ohs_site1.example.com`. The property name of the HTTP port is `HttpListenPort`, and the property name of the HTTPS port is `HttpSslListenPort`.

3. Change the listening port with the `beectl modify_property` command with the appropriate listening component and property name. The following example changes the HTTP port to 80:

```
beectl modify_property
--component ohs_site1.example.com --name HttpListenPort --value 80
```

4. Change the HTTP (or HTTPS) port number in the `_VIRTUAL_SERVER` component. The following example changes the HTTP port number. (Use the property name `HttpSslPort` to change the HTTPS port number):

```
beectl modify_property --component _VIRTUAL_SERVER --name HttpPort --value 80
```

5. Activate configuration and commit changes:

```
beectl activate_configuration
beectl modify_local_configuration_files
```

Note: The `beectl modify_local_configuration_files` command will ask you to run this command on all your other instances. **Do not run this command on all your other instances at this time.** For each instance, perform steps 1 to 3 before running the `beectl modify_local_configuration_files` command.

Changing Other Ports

Follow these steps to change ports of Oracle Beehive-specific protocols such as BTP/BTPS, IMAP/IMAPS and SMTP.

Note: If you are changing e-mail port numbers, refer to "[Configuring E-Mail with SSL](#)" for additional information about changing the ports that SMTP and IMAP listen to.

1. As the `root` user, change the owner of `hasbind` to the `root` user, set its user ID to `root`, and give all users permission to execute it:

```
chown root <Oracle home>/beehive/bin/hasbind
chmod a+x <Oracle home>/beehive/bin/hasbind
chmod a+s <Oracle home>/beehive/bin/hasbind
```

Note: For `hasbind` to work, ensure that `root` only has write permission to the `/etc` directory, while all other users and groups have only read permission.

2. As the `root` user, create a text file named `/etc/cap.ora` (that contains the following, where `instuser` is the user who installed Oracle Beehive:

```
+user instuser: bind port 80
```

Ensure the owner and user ID of `cap.ora` is `root`, give `root` read and write access to it, and all other users only read access:

```
chown root /etc/cap.ora
chmod 644 /etc/cap.ora
```

3. Determine the name of the listening component and property name of the port you want to change. Run the `beectl list_ports` command to list all available ports (the `--format` option is optional):

```
beectl list_ports --format xml
```

```
...
<row>
```

```

    <column name="Protocol">BTP</column>
    <column name="Listening Port">21401</column>
    <column name="Virtual Port">21401</column>
    <column name="Defining Component">SitewideBtiConfiguration</column>
    <column name="Property Name">ClientPort</column>
    <column name="Listening Component">BTI_site1.example.com</column>
</row>
<row>
    <column name="Protocol">BTPS</column>
    <column name="Listening Port">21451</column>
    <column name="Virtual Port">21451</column>
    <column name="Defining Component">SitewideBtiConfiguration</column>
    <column name="Property Name">SecureClientPort</column>
    <column name="Listening Component">BTI_site1.example.com</column>
</row>

```

In the previous example, the BTP (and BTPS) listening component is `BTI_site1.example.com`. The property name of the BTP port is `ClientPort`, and the property name of the BTPS port is `SecureClientPort`.

4. Change the listening port with the `beectl modify_property` command with the appropriate listening component and property name. The following example changes the BTP port:

```

beectl modify_property
  --component BTI_site1.example.com --name ClientPort --value <new port number>

```

5. Change the appropriate property in the `_VIRTUAL_SERVER` component to the new port number. The following example changes the BTI port to the new port number (to list the properties of `_VIRTUAL_SERVER`, run the command `beectl list_properties --component _VIRTUAL_SERVER`):

```

beectl modify_property
  --component _VIRTUAL_SERVER
  --name BtiClientPort
  --value <new port number>

```

6. Activate configuration and commit changes:

```

beectl activate_configuration
beectl modify_local_configuration_files

```

Note: The `beectl modify_local_configuration_files` command will ask you to run this command on all your other instances. **Do not run this command on all your other instances at this time.** For each instance, perform steps 1 to 3 before running the `beectl modify_local_configuration_files` command.

Enabling BPEL Workflows to Use New Port Number

If you changed the HTTP port number that Oracle Beehive listens to, follow these directions to enable BPEL workflows to use this new port number:

1. Modify the value element of the following properties in the file `<Oracle home>/bpel/system/config/collaxa-config.xml` to reflect the new HTTP port number:
 - `clusterName`
 - `soapCallbackUrl`

- soapServerUrl

The following is an example:

```
<property id="clusterName">
  <name>Cluster Id</name>
  <value>www.example.com:80</value>
</property>

<property id="soapCallbackUrl">
  <name>BPEL soap callback URL</name>
  <value>http://www.example.com:80</value>
</property>

<property id="soapServerUrl">
  <name>BPEL soap server URL</name>
  <value>http://www.example.com:80</value>
</property>
```

2. Restart the oc4j_soa component:

```
beectl restart --component oc4j_soa_instance1.example.com
```

Retrieve the component identifier of oc4j_soa with the `beectl status` command.

3. Re-deploy the SerialWorkflow and ParallelVoteWorkflow sample workflows by copying the jar files from `<Oracle home>/beehive/seed/workflows` into `<Oracle home>/bpel/domains/default/deploy`. Since the BPEL Process Manager is in auto-deploy mode, it will redeploy these processes after a few minutes.

Opening Ports Required by Oracle Beehive for FTP

If your Oracle Beehive instance is behind a firewall, ensure that the following ports are open in your firewall to enable FTP connections:

Table 12–1 Ports Required Open

Default Port Number	Property Name	Description
2121	Port	The FTP server listens at this port to accept new connections from clients
2120	DefaultDataPort	Port used by the FTP server in PORT (ACTIVE) mode to make active connections (outbound connections) to clients
12121	DataConnectionPort	In PASSIVE mode, the FTP server listens on a range of ports starting from DataConnectionPort to create new data connections, which will be used for data transfer operations. The property MaxDCPortCount indicates the number of ports in the range. For example, if the value of DataConnectionPort is 12121 and MaxDCPortCount is 10, then ensure that all the ports from 12121 to 12130 (inclusively) are open in your firewall.

You may change the value of these ports as well as the value of MaxDCPortCount.

To retrieve a list of all available ports, call the `beectl list_ports` command (the `--format` option is optional):

beectl list_ports --format xml

```
...
<row>
  <column name="Protocol">FTP</column>
  <column name="Listening Port">2121</column>
  <column name="Virtual Port">2121</column>
  <column name="Defining Component">_FtpService</column>
  <column name="Property Name">Port</column>
  <column name="Listening Component">
    BTI_instance1.example.com</column>
</row>
<row>
  <column name="Protocol"></column>
  <column name="Listening Port">2120</column>
  <column name="Virtual Port"></column>
  <column name="Defining Component">_FtpService</column>
  <column name="Property Name">DefaultDataPort</column>
  <column name="Listening Component">
    BTI_instance1.example.com</column>
</row>
<row>
  <column name="Protocol"></column>
  <column name="Listening Port">12121</column>
  <column name="Virtual Port"></column>
  <column name="Defining Component">_FtpService</column>
  <column name="Property Name">DataConnectionPort</column>
  <column name="Listening Component">
    BTI_instance1.example.com</column>
</row>
...
```

To retrieve a list of all the properties for the FTP service, call the `beectl list_properties` command:

beectl list_properties --component _FtpService

Property Name	Property Value
Port	2121
DataConnectionPort	12121
ChannelSecurity	PLAIN_AND_TLS
MaxDCPortCount	200
DefaultCommandCharacterSet	UTF-8
SessionTimeout	900000
UploadBufferSize	390144
DMSInstrumentation	false
FtpServerEnabled	true
Site	_CURRENT_SITE
ServiceInstances	instance_ftp_BEEAPP_instance1.example.com
ServiceApplication	svcapp_ftp
Alias	_FtpService

To change a property, call the `beectl modify_property` command:

```
beectl modify_property
  --component _FtpService
```

```
--name Port  
--value <new port value>
```

To save the your changes and activate them, call the following commands:

```
beectl activate_configuration  
beectl modify_local_configuration_files
```

Note: The `beectl modify_local_configuration_files` command will ask you to run this command on all your other instances. **Do not run this command on all your other instances at this time.** For each instance, make your desired changes to the Port property and run `beectl activate_configuration` before running the `beectl modify_local_configuration_files` command.

Configuring DMZ Instances

Refer to "[Configuring Oracle Beehive Demilitarized Zone Instances](#)" for instructions on how to configure DMZ instances and Oracle Wallet with DMZ instances.

Integrating and Synchronizing LDAP with Oracle Beehive

Oracle Beehive user data may be mastered in Oracle Beehive User Directory Service (UDS) or in an external LDAP-based directory, such as Oracle Internet Directory. "Mastered" means that a master source is used as the point of reference to determine the correct value for any user account attribute, and this source is used for making any changes to any account details.

After you have installed and configured Oracle Beehive, you may synchronize your external LDAP-based directory with UDS so that you may continue to manage your users and groups through your LDAP server. For more information, refer to "[Integrating and Synchronizing LDAP with Oracle Beehive](#)".

Configuring SSL

Refer to "[Configuring SSL](#)" for information about configuring SSL with Oracle Beehive and DMZ instances.

Configuring TLS

Some services, such as XMPP, require a Transport Layer Security (TLS) encrypted communication channel. Use Oracle Wallet to provide this.

To configure Oracle Beehive with Oracle Wallet so that clients may access it with a TLS connection, refer to "[Configuring TLS with Oracle Wallet](#)".

Configuring SSL for LDAP Integration

After configuring LDAP and Oracle Wallet with Oracle Beehive, you may configure Oracle Beehive authentication with Certificate Authority verification. For more information, refer to "[Configuring SSL for LDAP Integration](#)".

Enabling AJPS

To enable secure Apache JServ Protocol (AJPS), which is used instead of HTTP for communication between Oracle HTTP Server and OC4J, refer to ["Enabling AJPS"](#).

Configuring Oracle Beehive E-mail

The default installation of Oracle Beehive includes a basic configuration of e-mail services. This configuration does not include any filtering, dispatch rules, relay configuration, attachment blocking, or other common settings. For detailed configuration options and procedures, refer to "Managing Oracle Beehive E-Mail" in *Oracle Beehive Administrator's Guide*.

To secure Oracle Beehive e-mail with SSL, refer to ["Configuring E-Mail with SSL"](#). This involves configuring Oracle Beehive and your e-mail client.

Configuring Oracle Secure Enterprise Search

Oracle Secure Enterprise Search 10g is a stand-alone enterprise search solution. It incorporates best-in-class indexing crawling and security capabilities to create a reliable and comprehensive search solution for any organization. Refer to ["Configuring Oracle Secure Enterprise Search"](#).

Configuring Oracle Single Sign-On

You may register Oracle Beehive as a partner application with Oracle Single Sign-On, which means that you may delegate the authentication function to the single sign-on server. Refer to ["Configuring Single Sign-On with Oracle Beehive"](#).

Configuring External Oracle BPEL Process Manager with Oracle Beehive

If you have an existing instance of Oracle BPEL Process Manager, you may configure Oracle Beehive to use it instead of the one that is included with Oracle Beehive. Refer to ["Configuring External Oracle BPEL Process Manager with Oracle Beehive"](#).

Installing Oracle Beehive Integration for Outlook

Oracle Beehive Integration for Outlook (OBIO) extends the functionality of Microsoft Outlook by providing Outlook users with unified access to Oracle Beehive-based collaborative features and data in a familiar environment.

OBIO is supported in Windows operating systems with the following versions of Microsoft Outlook:

- Microsoft Outlook 2000
- Microsoft Outlook 2002
- Microsoft Outlook 2003
- Microsoft Outlook 2007

OBIO requires installation on the computers of individual users. For information about installing OBIO, refer to ["Installing Oracle Beehive Integration for Outlook"](#).

Configuring Oracle Beehive Integration for Zimbra

Oracle Beehive Integration for Zimbra is available for Oracle Beehive Release 1 (1.3) and later.

Oracle Beehive Integration for Zimbra is installed in an Oracle home separate from Oracle Beehive. As a result, Oracle Beehive Integration for Zimbra acts like an application tier and should be configured as such.

If you have not configured TLS or SSL for your Oracle Beehive deployment, you do not need to perform any post-installation steps; simply install Oracle Beehive Integration for Zimbra as described in ["Installing Oracle Beehive Integration for Zimbra"](#).

If you have configured TLS or SSL for your Oracle Beehive deployment, follow these steps to install Oracle Beehive Integration for Zimbra (these steps are similar to ["Configuring SSL with Real Certificates During Installation of Oracle Beehive"](#)):

1. If you have configured SSL for your Oracle Beehive deployment, ensure that all your Oracle Beehive application tiers are configured for SSL, including DMZ instances. Refer to ["Configuring SSL"](#).
2. Perform a software only install for Oracle Beehive Integration for Zimbra.
3. Replace `orapki` and Oracle Wallet Manager (`owm`) binaries of Oracle Beehive Integration for Zimbra with those from your Oracle Beehive instance. Create a new wallet located in `<Oracle Beehive Integration for Zimbra home>/Apache/Apache/conf/ssl.wlt/default`. Refer to ["Step 1: Creating Auto-Login Oracle Wallet"](#) for directions to create a wallet.
4. If you are using real (CA-signed) certificates for your application tier wallets, perform this step.

Remove the test certificates using Oracle Wallet Manager from the wallets in `<Oracle Beehive Integration for Zimbra home>/opmn/conf/ssl.wlt/default` and `<Oracle Beehive Integration for Zimbra home>/Apache/Apache/conf/ssl.wlt/default`, if any. The order of removal should be (1) user certificate, (2) certificate request, and (3) trusted certificate. Refer to ["Creating CA-Signed Certificate and Importing it into Wallet"](#) for information about CA-signed certificates.

5. Run the Config Wizard for Oracle Beehive Integration for Zimbra and complete the configuration.
6. Configure TLS for Oracle Beehive Integration for Zimbra. Perform ["Step 2: Configuring Oracle Beehive Instance to Use Oracle Wallet"](#). (Note that you already created an auto-login wallet for Oracle Beehive Integration for Zimbra in a previous step.)
7. If you are using real (CA-signed) certificates for your application tier wallets, perform this step.

Remove the test certificates using Oracle Wallet Manager from the wallets in `<Oracle Beehive Integration for Zimbra home>/Apache/Apache/conf/ssl.wlt/default`, if any.

Note: Oracle Beehive Integration for Zimbra installation will fail if the default test certificates are still in the wallets of your Oracle Beehive instances. Configure TLS on the failed Oracle Beehive Integration for Zimbra home, as described in "[Configuring TLS with Oracle Wallet](#)", and click **Retry** from the Install Wizard.

Cloning Oracle Beehive

Cloning is the process of copying an existing installation to a different location while preserving its configuration. For more information, refer to "[Cloning Oracle Beehive Application Tiers and Sites](#)".

Performing Oracle Beehive Administration Console Post-Installation Procedures

Oracle Beehive Administration Console is available for Oracle Beehive Release 1 (1.3) and later.

If you have installed Oracle Beehive Administration Console, perform one or more of the procedures described in "[Oracle Beehive Administration Console Post-Installation Procedures](#)", depending on your security requirements or any other issue particular to your deployment.

Configuring Oracle Beehive Demilitarized Zone Instances

This module describes how to configure Oracle Beehive DMZ instances. It involves the following steps:

- [Step A: Configuring Oracle Wallet with Oracle Beehive DMZ Instances](#)
- [Step B: Configuring Oracle Beehive DMZ Instances](#)

Note: If you want to configure your DMZ instances with SSL, follow the directions in "[Configuring SSL](#)". These steps involve configuring your entire Oracle Beehive deployment for SSL.

Step A: Configuring Oracle Wallet with Oracle Beehive DMZ Instances

Follow these steps on all DMZ instances. Refer to "[Configuring TLS with Oracle Wallet](#)" for more information about Oracle Wallet.

1. Set the ORACLE_HOME environment variable to the Oracle home of the DMZ instance.
2. Run the following command to create a wallet:

```
<Oracle home of DMZ instance>/bin/orapki wallet create
-wallet <Oracle home of DMZ instance>/Apache/Apache/conf/ssl.wlt/default
-auto_login
-pwd <new password for wallet>
```

Note: The default Oracle Wallet directory is *<Oracle home of DMZ instance>/Apache/Apache/conf/ssl/default*. Alternatively, you may create a new Oracle Wallet in a different directory (by changing the value of the `-wallet` option..

3. Edit the file *<Oracle home of DMZ instance>/beehive/conf/bti.properties* and change the value of `WalletDir` to the wallet directory you specified in the previous step:

```
WalletDir=
<Oracle home of DMZ instance>/Apache/Apache/conf/ssl.wlt/default
```

4. Edit the file *<Oracle home of DMZ instance>/opmn/conf/opmn.xml* and change the value of `wallet-file` to the wallet directory you specified in Step 2:

```
<notification-server interface="ipv4">
  <!-- ... -->
  <ssl enabled="true"
    wallet-file="<Oracle home of DMZ instance>
      /Apache/Apache/conf/ssl.wlt/default"/>
```

- Restart the DMZ instance:

```
opmnctl stopall
opmnctl startall
```

Step B: Configuring Oracle Beehive DMZ Instances

- If you have not already done so, configure Oracle Wallet with your Oracle Beehive DMZ instance. Refer to ["Step A: Configuring Oracle Wallet with Oracle Beehive DMZ Instances"](#).
- If you specifically do not want to use SSL, then perform this step. Otherwise ignore this step.

Edit the file `<Oracle home of DMZ instance>/opmn/conf/opmn.xml` as follows in all Oracle Beehive DMZ instances. In the `<ssl>` element, set the `enable` attribute to `false`:

```
<ssl enabled="false"
  wallet-file="<ORACLE_HOME>/opmn/conf/ssl.wlt/default"/>
```

- Update the file `<Oracle home of DMZ instance>/opmn/conf/opmn.xml` as follows in all Oracle Beehive DMZ instances to add the topology definition for all Oracle Beehive (non-DMZ) application tiers that are part of this configuration:

```
<notification-server>
  <!-- ... -->
  <topology>
    <nodes list="
      <Application tier 1 host name>:<OPMN remote port of application tier 1>,
      <Application tier 2 host name>:<OPMN remote port of application tier 2>,
      ...
    "/>
  </topology>
</notification-server>
```

For example, if you have two Oracle Beehive application tiers with host names `example1.com` and `example2.com` and OPMN remote port numbers 6200 and 6300, respectively, add the following topology definition to the `opmn.xml` file in your DMZ instances:

```
<notification-server>
  <!-- ... -->
  <topology>
    <nodes list="example1.com:6200,example2.com:6300"/>
  </topology>
</notification-server>
```

Tip: Retrieve the OPMN port of an Oracle Beehive application tier from the file `<Oracle Beehive home>/opmn/conf/opmn.xml`. The OPMN port is specified by the `remote` attribute of the `port` element in the `notification-server` element.

4. Restart each DMZ instance whose `opmn.xml` file you changed with the `<Oracle home of DMZ instance>/opmn/bin/opmnctl` command:

```
opmnctl stopall  
opmnctl startall
```
5. From every DMZ instance, retrieve the following values from the specified files:
 - `<Oracle home of DMZ instance>/opmn/conf/opmn.xml`:
 - `opmn_request_port`
 - `opmn_remote_port`
 - `<Oracle home of DMZ instance>/beehive/conf/bti.properties`:
 - `bti_server_port`
 - `bti_unique_id` (PersistentId)
 - `NumberOfClientWorkers`
6. Open the following ports in your firewall:
 - OPMN ports between your DMZ instances and non-DMZ instances
 - AJP ports between your intranet and your DMZ instances: This is required for connectivity between Oracle HTTP Server (OHS) and OC4J. By default, Oracle Beehive OC4J instances listens on port numbers in the range 12501-12600. Oracle Beehive OC4J instances will try to listen on the lowest port number in this range.

Consequently, you only have to open the range of ports 12501-12504. Ensure that no other applications on Oracle Beehive servers occupy this range. As long as these ports are not occupied, Oracle Beehive OC4J instances will listen on this range of ports.
 - If you have installed Oracle Beehive Integration for Zimbra, open two additional ports (12505-12506).

Notes: You do not need to open any OPMN ports between your DMZ instances and any Oracle RAC database host(s).

The AJP port number range is defined in the `AjpPortMinValue` and `AjpPortMaxValue` parameters in each of Oracle Beehive's managed OC4J components:

```
beectl list_components --type ManagedOc4j
```

```
-----+-----
Component type | Component identifier
-----+-----
ManagedOc4j   | BEEAPP_site.example.com
-----+-----
```

...

```
beectl list_properties --component BEEAPP_site.example.com
```

```
-----+-----
Property name   | Property value
-----+-----
AdminPassword   | [Protected Value]
-----+-----
AdminUsername   | oc4jadmin
-----+-----
AjpPortMaxValue | 12600
-----+-----
AjpPortMinValue | 12501
-----+-----
```

...

You may open a range of ports other than 12501-12504 (or 12501-12506 if you have installed Oracle Beehive Integration for Zimbra); however, you must make the necessary changes in the Oracle Beehive configuration.

7. On any Oracle Beehive non-DMZ instance, **but not on a DMZ instance**, run the following commands:

- Add the first DMZ instance to the non-DMZ instance:

```
beectl add_dmz_home_instance
  --hostname <Host name of first DMZ instance>
  --oracle_home <Oracle home of DMZ instance>
  --opmn_request_port <OPMN request port of first DMZ instance>
  --opmn_remote_port <OPMN remote port of first DMZ instance>
  --bti_server_port <BTI server port of first DMZ instance>
  --bti_unique_id <BTI unique ID of first DMZ instance>
  --no_of_client_workers <NumberOfClientWorkers from bti.properties>
```

- For the next and all other DMZ instances, run the `beectl add_dmz_home_instance` command (on the same non-DMZ instance) with the appropriate parameters you retrieved from step 2.

8. From the same non-DMZ instance, run the `beectl activate_configuration` command:

```
beectl activate_configuration
```

9. Run the following command on all the other non-DMZ instances:

```
beectl modify_local_configuration_files --log_level FINEST
```

Troubleshooting DMZ Configuration

If you receive an HTTP 404 error when accessing an HTTP resource from a DMZ instance, verify that you have properly configured your DMZ instances. Do this by running the command `<Oracle home>/opmn/bin/opmnctl @cluster status` from any Oracle Beehive instance. If you have properly configured your DMZ instances, then this command will display all your application tiers in your site.

Integrating and Synchronizing LDAP with Oracle Beehive

Oracle Beehive user data may be mastered by the Oracle Beehive User Directory Service (UDS) or an external LDAP-based directory. "Mastered" means that a master source is used as the point of reference to determine the correct value for any user account attribute, and this source is used for making any changes to any account details.

This module describes how to integrate and synchronize UDS with an external LDAP-based directory, such as Oracle Internet Directory, so that all user data is mastered by the LDAP-based directory.

If UDS is synchronized with an external LDAP server, it will contact the LDAP server at regular intervals for all records that were changed. UDS will update its records accordingly. You may change the frequency that UDS contacts the LDAP server.

If you make a change in UDS, it will not update the LDAP server with which it is synchronized.

Note: It is not necessary to master all user account attributes in an LDAP server; some attributes may be mastered in LDAP and others in UDS. However, all users that need to authenticate or login must be mastered in the same place.

The process of synchronizing an LDAP server with UDS involves creating the following files:

- **LDAP mapping profile:** This is an XML file that contains LDAP server settings and specifies how to convert entries in your LDAP server to Oracle Beehive users and groups. This involves specifying attribute mappings between those defined in your LDAP server and those used by Oracle Beehive.
- **User file:** This is an XML file that represents all the users you want to synchronize in a format specified by Oracle Beehive. To create this, you use the LDAP mapping profile you created and the `beectl download_ldap_user_data` command.

This module covers the following topics:

- [How Synchronization Works](#)
- [Requirements](#)
- [Synchronizing LDAP with Oracle Beehive User Directory Service](#)
- [Configuring Authentication Service to Use LDAP Server](#)

- [Configuring Digest Authentication](#)
- [Oracle Internet Directory Considerations](#)
- [Changing LDAP Administrator's Password](#)
- [Active Directory Considerations](#)
- [Troubleshooting General LDAP Synchronization Issues](#)

How Synchronization Works

An LDAP server maintains a change log in which it stores incremental changes made to directory objects. It stores these changes sequentially based on the change log number.

UDS contacts the LDAP server at regular intervals (whose duration is determined by the LDAP mapping profile) and requests the latest change log number and all records that were changed since the last stored change log number.

Note: If any change is made to the LDAP server that alters the change log number, such as restoring the LDAP server from a backup or switching to a cloned instance, you must export all users and groups from LDAP and import them into UDS again as described in "[Step 3: Loading Users and Groups](#)".

Because any LDAP server may be synchronized with UDS, you must provide UDS with an LDAP mapping profile. This file specifies how to convert entries in your LDAP server to Oracle Beehive users and groups and which entries to synchronize.

Requirements

Oracle Beehive Release 1 supports Oracle Internet Directory, Microsoft Active Directory, IBM Tivoli Directory Server, and Sun Directory Server.

Synchronizing LDAP with Oracle Beehive User Directory Service

Synchronizing LDAP with UDS consists of the following steps:

- [Step 1: Creating an LDAP Mapping Profile](#)
- [Step 2: Enabling Synchronization](#)
- [Step 3: Loading Users and Groups](#)

Notes: You will need the user name and password of a user of your LDAP server who has access to the following:

- Attributes in the Directory Information Tree (DIT)
- Change logs

This user does not need write access to your LDAP server.

The steps in this module will use the user `cn=orcladmin`.

These steps use Oracle Internet Directory as the LDAP server to synchronize. For information specific to Active Directory, refer to the section "[Active Directory Considerations](#)".

This section also covers these topics:

- [Controlling How Often UDS Contacts the LDAP Server](#)
- [Retrieving Information About the LDAP Server](#)

Step 1: Creating an LDAP Mapping Profile

The LDAP mapping profile is an XML file that tells UDS the following information:

- Which LDAP entries should be synchronized
- How to treat entries with specific attributes or domain names (DNs) (for example, whether to map them as `ENTERPRISE_USER`, `EXTENDED_ENTERPRISE_USER`, or `EXTERNAL_PERSON`)
- How to map the attributes of each user type to Oracle Beehive attributes.

Creating the LDAP mapping profile consists of the following steps:

- [Step A: Creating an LDAP Mapping Profile from a Template](#)
- [Step B: Renaming the Profile](#)
- [Step C: Specifying LDAP Server Settings](#)
- [Step D: Providing Mapping Details for Each User Type and Static Group](#)
- [Step E: Providing Scope and Membership Mapping Information](#)
- [Step F: Providing Attribute Mapping for Each User Type and Static Group](#)
- [Step G: Adding Profile to Oracle Beehive](#)

Step A: Creating an LDAP Mapping Profile from a Template

Navigate to the directory `<Oracle home>/beehive/templates/uds`. It contains LDAP mapping profile templates for the following LDAP servers. These templates must be edited and customized depending on how your LDAP directory is configured and structured:

Table 14–1 LDAP Mapping Profile Templates

File Name	Directory Type
<code>adprofile_template.xml</code>	Microsoft Active Directory
<code>ibmprofile_template.xml</code>	IBM Tivoli Directory Server
<code>oidprofile_template.xml</code>	Oracle Internet Directory

Table 14–1 (Cont.) LDAP Mapping Profile Templates

File Name	Directory Type
sunprofile_template.xml	Sun Directory Server

Depending on your LDAP server, copy one of these files to another location, such as your home directory. Edit this file to create your LDAP mapping profile.

These steps use `oidprofile_template.xml`.

Step B: Renaming the Profile

Rename the profile in the `<profile_name>` tag. The following is an excerpt from an LDAP mapping profile:

```
<profile>
  <profile_name>my_profile</profile_name>
  <!-- ... -->
</profile>
```

Step C: Specifying LDAP Server Settings

Enter the LDAP server's host and port, administrator's username and password (which must be obfuscated), and the users and groups base search. The following is an excerpt from an LDAP mapping profile:

```
<ldap_server>
  <host>www.ldapserver.com</host>
  <port>389</port>
  <!-- <ssl_port>636</ssl_port> -->
  <connection_timeout>
    120
    <!-- This is the default value, in seconds -->
  </connection_timeout>
  <ldap_user_name>cn=orcladmin</ldap_user_name>
  <!-- obfuscated password -->
  <ldap_user_password>
    fCgF4UPWg+Vm7IkSBSY07NOSkJ2XXTYRwGynrIM0mx/CHQF4W58Mab0izRX6Bxb6
  </ldap_user_password>
  <user_search_base>dc=oracle,dc=com</user_search_base>
  <group_search_base>
    cn=groups,dc=us,dc=oracle,dc=com
  </group_search_base>
  <primary_authentication_attribute>uid</primary_authentication_attribute>
  <!-- The primary authentication attribute is required only
    for DEFAULT profile -->
  <user_search_base_for_sync>
    <!-- Do not use; a value specified here will
      override the value specified in user_search_base. -->
  </user_search_base_for_sync>
  <digest_authentication>
    <!-- Corresponds to the DigestAuthentication
      property of the component _CURRENT_SITE:LdapServer.
      This property can have multiple digest_authentication_attribute
      values -->
    <digest_authentication_attribute>
      <!-- An attribute from the user object (in the LDAP directory)
        that is required for digest authentication -->
    </digest_authentication_attribute>
  </digest_authentication>
```



```
</ldap_server>
```

In this excerpt, only users under `dc=oracle,dc=com` will be mapped to Oracle Beehive users. Similarly, only groups under `cn=groups,dc=us,dc=oracle,dc=com` will be mapped to Oracle Beehive groups.

The `<connection_timeout>` element is used by UDS to establish a connection to an external directory. If UDS cannot establish a connection within the number of seconds specified in this element, it aborts the connection attempt. The default value is 120 seconds. This element is available for Oracle Beehive Release 1 (1.3) and later.

For more information about the `DigestAuthentication` property, refer to ["Configuring Digest Authentication"](#).

Notes: The LDAP user specified in `<ldap_user_name>` must have access to the change logs. If you later update the profile with a different LDAP user, then UDS will be synchronized with the state of the LDAP server corresponding to the latest change log number.

To obfuscate the LDAP administrator's password, use the `beectl obfuscate` command:

```
beectl obfuscate --expiration_time_in_minutes 0
Enter value for password:
Confirm value of password:
Successfully obfuscated the string.
fCgF4UPWg+Vm7IkSBSY07NOSkJ2XXTYRwGynrIM0mx/CHQF4W58Mab0izRX6Bxb6
```

Other `beectl` commands require obfuscated passwords. Use the same command to obfuscate them.

Step D: Providing Mapping Details for Each User Type and Static Group

Provide the mapping details for each user type and static group. The following is an excerpt from an LDAP mapping profile:

```
<user_type_map>
  <user_type_map_entry>
    <source_field_type>DN</source_field_type>
    <source_field_value>
      cn=users,dc=partners,dc=oracle,dc=com
    </source_field_value>
    <user_type>EXTENDED_ENTERPRISE_USER</user_type>
  </user_type_map_entry>
  <user_type_map_entry>
    <source_field_type>DN</source_field_type>
    <source_field_value>
      cn=users,dc=us,dc=oracle,dc=com
    </source_field_value>
    <user_type>ENTERPRISE_USER</user_type>
  </user_type_map_entry>
</user_type_map>

<group_type_map>
  <group_type_map_entry>
    <source_field_type>DN</source_field_type>
    <source_field_value>
      cn=groups,dc=us,dc=oracle,dc=com
    </source_field_value>
    <group_type>STATIC_GROUP</group_type>
  </group_type_map_entry>
```

```
</group_type_map>
```

This excerpt maps the following entries:

- A user that is under the DN specified in `<user_search_base>` (in this example, it is `dc=oracle,dc=com`) and whose DN contains `cn=users,dc=partners,dc=oracle,dc=com` will be mapped to `EXTENDED_ENTERPRISE_USER`.
- An entry that is under the DN specified `<user_search_base>` and whose DN contains `cn=users,dc=us,dc=oracle,dc=com` will be mapped to `ENTERPRISE_USER`.

Note: Users of type `EXTENDED_ENTERPRISE_USER` must be specified before users of type `ENTERPRISE_USER`.

Exclusion and Inclusion Consider the following example:

```
<user_type_map>
  <user_type_map_entry>
    <source_field_name>UserStatus</source_field_name>
    <source_field_type>ATTRIBUTE</source_field_type>
    <source_field_value>true</source_field_value>
    <user_type>ENTERPRISE_USER</user_type>
  </user_type_map_entry>
</user_type_map>
```

In this example, a user (created in your LDAP directory) whose `UserStatus` attribute is set to `true` will be mapped to `ENTERPRISE_USER`.

However, if `UserStatus` is changed to any value other than `true` or nullified, then UDS synchronization will set the user's status as `DISABLED` in Oracle Beehive because the user no longer satisfies the condition specified in this `<user_type_map>`.

If `UserStatus` is changed back to `true`, then UDS synchronization will set the user's status as `ENABLED` in Oracle Beehive.

If a user in LDAP is deleted, UDS synchronization will set the user's status as `MARKED_FOR_DELETE` in Oracle Beehive.

Step E: Providing Scope and Membership Mapping Information

Provide community mapping information. Enter this information in an `<scope_type_map>` element. Users specified in a `<scope_type_map>` will be added to, or *scoped* within, the community (organization or enterprise) specified in the same element. A user may only be scoped within a single community.

You may optionally specify a `<membership_type_map>` element. Users specified in this element will be scoped within the community (organization or enterprise) specified by `<scope_type_map>`. In addition, users will become a member of the community specified in the `<membership_type_map>` element. A user may be a member of zero or more communities.

The following is an excerpt from an LDAP mapping profile.

```
<scope_type_map>

  <scope_type_map_entry>
    <source_field_name>OU</source_field_value>
    <source_field_type>ATTRIBUTE</source_field_type>
```

```

<source_field_value>Alpha</source_field_value>
<scope>
  <name>My_Organization</name>
  <identifier>orgn=My_Organization,enpr=My_Enterprise</identifier>
</scope>
</scope_type_map_entry>

<scope_type_map_entry>
<source_field_type>DN</source_field_type>
<source_field_value>dc=us,dc=example,dc=com</source_field_value>
<scope>
  <name>Entr1</name>
  <identifier>enpr=Oracle</identifier>
</scope>
<membership_type_map>
  <membership_type_map_entry>
    <source_field_type>DN</source_field_type>
    <source_field_value>
      dc=external,dc=us,dc=example,dc=com</source_field_value>
    <name>My_Organization</name>
    <identifier>orgn=My_Organization,enpr=My_Enterprise</identifier>
  </membership_type_map_entry>
</membership_type_map>
</scope_type_map_entry>
<scope_type_map_entry>
</scope_type_map>

```

This excerpt maps the following entries:

- A user that is under the attribute OU=Alpha will be scoped within the organization My_Organization.
- A user that is under the DN dc=us, dc=oracle, dc=com will be scoped within the enterprise My_Enterprise. A user that is under the attribute dc=external, dc=us, dc=example, dc=com will be scoped within the same enterprise (My_Enterprise). The same user will be a member of the organization My_Organization.

Tips: To retrieve the identifier for an enterprise, call the following `beectl` command:

```
beectl list_enterprises
```

```

-----
| Enterprise Name      | Identifier                |
-----
| MyEnterprise         | enpr=My_Enterprise      |
-----

```

To retrieve the identifier for an organization, call the following `beectl` command:

```
beectl list_organizations --scope enpr=My_Enterprise
```

```

Organization name:    My_Organization
Description:         Unknown
Identifier:           orgn=My_Organization,enpr=My_Enterprise
...

```

Step F: Providing Attribute Mapping for Each User Type and Static Group

Provide the attribute mapping for each user type and static group. The following is an excerpt from an LDAP mapping profile:

```
<directory_attribute_map>
  <directory_attribute_map_entry>
    <source_object>ENTERPRISE_USER</source_object>
    <AttributeMap>
      <Field>
        <source_attribute>givenname</source_attribute>
        <target_attribute>GIVENNAME</target_attribute>
        <target_attribute_type>ATTRIBUTE</target_attribute_type>
      </Field>
      <Field>
        <source_attribute>sn</source_attribute>
        <target_attribute>FAMILYNAME</target_attribute>
        <target_attribute_type>ATTRIBUTE</target_attribute_type>
      </Field>
    </AttributeMap>
  </directory_attribute_map_entry>

  <directory_attribute_map_entry>
    <source_object>EXTENDED_ENTERPRISE_USER</source_object>
    <AttributeMap>
      <Field>
        <source_attribute>givenname</source_attribute>
        <target_attribute>GIVENNAME</target_attribute>
        <target_attribute_type>ATTRIBUTE</target_attribute_type>
      </Field>
      <Field>
        <source_attribute>sn</source_attribute>
        <target_attribute>FAMILYNAME</target_attribute>
        <target_attribute_type>ATTRIBUTE</target_attribute_type>
      </Field>
    </AttributeMap>
  </directory_attribute_map_entry>

  <directory_attribute_map_entry>
    <source_object>STATIC_GROUP</source_object>
    <AttributeMap>
      <Field>
        <source_attribute>displayname</source_attribute>
        <target_attribute>NAME</target_attribute>
        <target_attribute_type>ATTRIBUTE</target_attribute_type>
      </Field>
    </AttributeMap>
  </directory_attribute_map_entry>
</directory_attribute_map>
```

In this excerpt, for each `ENTERPRISE_USER`, the `givenname` LDAP attribute will be mapped to the `GIVENNAME` attribute in Oracle Beehive. Similarly, for each `STATIC_GROUP`, the `displayname` LDAP attribute will be mapped to the `NAME` attribute in Oracle Beehive.

Mapping Active Directory Proxy Addresses An Active Directory user's entry contains an attribute named `proxyAddresses` that holds all the e-mail addresses of a particular user.

The following is an example of a `proxyAddresses` attribute:

```
proxyAddresses: smtp:rholmes@example.com ;
SMTP:Robert.Holmes@example.com ; MBX:0 ; X400:c=US ; p=Example ;
s=Holmes ; g=Robert ; RFAX: Holmes, Robert @
```

Consider the following example:

```
<Field>
  <source_attribute>proxyAddresses</source_attribute>
  <target_attribute>PROXY</target_attribute>
  <target_extended_attribute>MAILTO</target_extended_attribute>
  <target_attribute_type>ADDRESS</target_attribute_type>
  <source_special_handling>PROXY</source_special_handling>
</Field>
```

If the `<source_special_handling>` element is omitted, UDS synchronization will map the Active Directory `proxyAddresses` value of a user to the Oracle Beehive address type `PROXY` with the scheme `MAILTO`. This creates the following:

```
proxy1:mailto:smtp:rholmes@example.com
proxy2:mailto:smtp:robert.holmes@example.com
...
proxy8:mailto:RFAX: Holmes, Robert
```

If the `<source_special_handling>` element is included, Oracle Beehive will only synchronize values that start with `smtp:` and remove the text `smtp:.` As a result, the actual values in Oracle Beehive become the following:

```
proxy1:mailto:rholmes@example.com
proxy2:mailto:robert.holmes@example.com
```

Consequently, the `<source_special_handling>` element properly formats Active Directory e-mail addresses for Oracle Beehive and ignores values that start with `MBX`, `RFAX`, and other protocols not used by Oracle Beehive. This kind of mapping allows you to incorporate your legacy e-mail addresses into Oracle Beehive by synchronizing it with Active Directory's method of inbound mail lookup resolution.

Step G: Adding Profile to Oracle Beehive

Add the profile with the following `beectl` command:

```
beectl add_directory_profile --file oidprofile_template.xml
Successfully added 1 directory profiles.
```

The utility may return an error similar to the following:

```
Failed to add directory profiles. See the log file.
```

The log file is `<Oracle home>/beehive/logs/oc4j/BEEMGMT/log.txt`.

Note: To modify a directory profile, you may use the `beectl modify_directory_profile` command, delete the existing directory profile and then add the updated profile, or modify one or more properties of the existing directory profile with the `beectl modify_property` command. In either case, you do not have to restart the LDAP server.

This command is available for Oracle Beehive Release 1 (1.3) and later.

Directory Profile Validation When you add a directory profile, Oracle Beehive validates the following in the XML file:

1. LDAP credentials
2. <poll_interval>, <profile_flag>, and <directory_type>
3. The existence of <user_search_base> and <group_search_base> in your LDAP server
4. For <scope_type_map> and <membership_type_map>, the following are validated:
 - a. <source_field_type> (either DN or ATTRIBUTE)
 - b. <source_field_value> if <source_file_type> is DN (if <source_field_type> is ATTRIBUTE, then this validation is skipped)
 - c. Values defined in <identifier> are validated for their existence; if you have specified an invalid enterprise or organization identifier, then an appropriate error message is returned
5. For <user_type_map>, the following are validated:
 - a. <source_field_type> (either DN or ATTRIBUTE)
 - b. <source_field_value> if <source_file_type> is DN (if <source_field_type> is ATTRIBUTE, then this validation is skipped)
 - c. <user_type> (either ENTERPRISE_USER, EXTENDED_ENTERPRISE_USER, or EXTERNAL_PERSON)
6. For <group_type_map> the following are validated:
 - a. <source_field_type> (either DN or ATTRIBUTE)
 - b. <source_field_value> if <source_file_type> is DN (if <source_field_type> is ATTRIBUTE, then this validation is skipped)
 - c. <group_type> (only valid value is STATIC_GROUP)
7. For <directory_attribute_map> the following are validated:
 - a. <target_attribute>: If <source_object> is ENTERPRISE_USER or EXTENDED_ENTERPRISE_USER, attribute mappings for PRINCIPAL and FAMILYNAME target attributes must exist. If <source_object> is STATIC_GROUP, attribute mappings for NAME target attribute must exist. If <source_object> is EXTERNAL_PERSON, attribute mappings for FAMILYNAME target attribute must exist.
 - b. <target_attribute_type>
 - c. <target_extended_attribute>

Step 2: Enabling Synchronization

Enable the synchronization profile with the following commands: These commands enable a profile named `oidldapdirectoryprofile`:

```
beectl list_properties --component oidldapdirectoryprofile
beectl modify_property --component oidldapdirectoryprofile
                        --name ProfileState --value ENABLE
beectl activate_configuration
```

Note: Your users will not be able to login yet, even though they are provisioned. Your users will be able to login once you have completed the step described in "[Configuring Authentication Service to Use LDAP Server](#)".

Step 3: Loading Users and Groups

The following steps describe how to load all users and groups from the LDAP server to UDS:

1. Generate an XML file from the LDAP server based on the mapping profile you loaded into Oracle Beehive in the previous step. The following command will create a file named `UsersFromLDAP.xml` in your home directory based on the profile named `oidldapdirectoryprofile`:

```
beectl download_ldap_user_data
--file UsersFromLDAP.xml
--profile oidldapdirectoryprofile
```

Note: You do not need administrator privileges to the LDAP server in order to extract data from it. Therefore a normal user may run the `beectl download_ldap_user_data` command.

However, LDAP directories may impose a search limit for non-administrator users.

For example, if your OID server has 500 records, OID may impose a search limit of 200 records for non-administrator users. If you are a normal user, the `beectl download_ldap_user_data` command will return only 200 records. As a result, you will not be able to synchronize all your users.

Check your LDAP server documentation for maximum returned result limitations and how to manage them.

2. In a text editor, open the file you generated (`UsersFromLDAP.xml`), and check for the following:
 - `primary_principal` is mapped to the attribute your LDAP server is configured for authentication, for example, `sAMAccountName` for Active Directory or `uid` (by default) for Oracle Internet Directory, otherwise authentication will fail
 - Enterprise and organization identifiers are correct for your Oracle Beehive deployment and all the organizations already exist in Oracle Beehive
 - The element `familyname` is defined and contains a value for each user

Notes: If you receive many errors or inconsistencies in the generated XML file, delete it, correct the LDAP mapping profile, and recreate the generated XML file.

3. Add the users in the generated XML file to Oracle Beehive with the `beectl add_user` command:

```
beectl add_user --file UsersFromLDAP.xml
```

```
--ldapbootstrap
```

4. Make sure that the users were added successfully with the `beectl list_users` command:

```
beectl list_users
```

Controlling How Often UDS Contacts the LDAP Server

By default, UDS contacts the LDAP server's change log every 30 seconds for updates. You may change this interval in either of the following ways:

- In your LDAP mapping profile, change the value in the `<poll_interval>` tag.

The following is an excerpt from an LDAP mapping profile with an interval set to 15 seconds:

```
<profile>
  <profile_name>oidldapdirectoryprofile</profile_name>
  <poll_interval>15</poll_interval>
  <profile_state>DISABLE</profile_state>
  <profile_flag>DEFAULT</profile_flag>
  <directory_type>ORACLE_INTERNET_DIRECTORY</directory_type>
  <ldap_server>
  <!-- ... -->
</profile>
```

If you make any changes to your LDAP mapping profile, you will have to delete and recreate your existing profile.

Notes: The value of the `<profile_flag>` element may be `DEFAULT` or `NON_DEFAULT`.

A `DEFAULT` profile is the one used by both authentication and UDS. This profile stores and reads LDAP server information from the site.

For `NON_DEFAULT` profiles, the `LdapServer` object is stored within the profile itself.

In Oracle Beehive Release 1 (1.2) only, the `<profile>` element requires one more element, `<profile_mode>` to be set as `SYNC`:

```
<profile>
  ...
  <profile_state>DISABLE</profile_state>
  <profile_mode>SYNC</profile_mode>
```

- Use the `beectl modify_property` command. The following commands set the value of the property `PollInterval` to 15 seconds:

```
beectl list_properties --component oidldapdirectoryprofile
beectl modify_property
  --component oidldapdirectoryprofile
  --name PollInterval
  --value 15
beectl activate_configuration
```

Note: If the LDAP server's change log is cleaned up or purged more frequently than the UDS update frequency, data might be lost.

Retrieving Information About the LDAP Server

When you create a profile, Oracle Beehive creates an `LdapServer` configuration object. Use the `beectl list_properties` to get information about it:

```
beectl list_properties --component _CURRENT_SITE:LdapServer
```

Property Name	Property Value
LdapServerHostName	ldapserver.com
LdapServerPort	389
LdapServerSslPort	636
SslEnabled	false
LdapServerUser	cn=orcladmin
LdapServerPassword	[Protected Value]
SSLMode	0
UserSearchBase	cn=users,dc=us,dc=oracle,dc=com
UserSearchBaseForSync	
GroupSearchBase	cn=groups,dc=us,dc=oracle,dc=com
UserObjectClass	
GroupObjectClass	
PrimaryAuthenticationAttribute	uid
PrimaryAuthenticationCredential	not applicable
ProtocolAuthenticationAttribute	not applicable
ProtocolAuthenticationCredential	not applicable
VoiceAuthenticationAttribute	not applicable
VoiceAuthenticationCredential	not applicable
DirectoryType	ORACLE_INTERNET_DIRECTORY
Alias	OracleInternetDirectory

Notes: You may have multiple `LdapServer` objects in an Oracle Beehive deployment if you have configured more than one LDAP mapping profile. The Authentication Service uses the `LdapServer` object set at the site level, which is created by the UDS mapping profile.

The site level `LdapServer` object may not have all required properties for authentication.

The following table describes the properties of the `LdapServer` object:

Table 14–2 *LdapServer Properties*

Property Name	Required	Description
LdapServerHostName	Required	LDAP server host name
LdapServerPort	Required	LDAP server port for non-SSL connections
LdapServerSslPort	Required	LDAP server port for SSL connections
SslEnabled	Required	If set to true, only SSL connections are used
LdapServerUser	Required	LDAP server user with bind and search privileges. This user must be able to look up attributes for all LDAP users provisioned to use Oracle Beehive
LdapServerPassword	Required	Password for LdapServerUser

Table 14–2 (Cont.) LdapServer Properties

Property Name	Required	Description
SSLMode	Not used	Ignore this property
UserSearchBase	Required	User search base dn. The search scope is always subtree (recursive search).
GroupSearchBase	Required	Group search base dn. Search Scope is always subtree (recursive search).
UserObjectClass	Optional	Name of the user object class in the directory. This attribute is used to construct a search filter for the users. If this value is not specified, a default value is used, described in "Default UserObjectClass and GroupObjectClass Values".
GroupObjectClass	Optional	Name of the group object class in the directory. This attribute is used to construct a search filter for the groups. If this value is not specified, a default value is used, described in "Default UserObjectClass and GroupObjectClass Values".
PrimaryAuthenticationAttribute	Required	The name of the attribute the LDAP server uses to authenticate a user. For example, set this to uid for Oracle Internet Directory, or sAMAccountName for Active Directory.
PrimaryAuthenticationCredential	Not used	Ignore this property
ProtocolAuthenticationAttribute	Not used	Ignore this property
ProtocolAuthenticationCredential	Not used	Ignore this property
VoiceAuthenticationAttribute	Not used	Ignore this property
VoiceAuthenticationCredential	Not used	Ignore this property
DirectoryType	Required	Indicates which specific LDAP directory is being configured. Valid values are the following: ORACLE_INTERNET_DIRECTORY MICROSOFT_ACTIVE_DIRECTORY IBM_TIVOLI_DIRECTORY SUN_ONE_DIRECTORY
Alias	Optional	Alias for this LdapServer configuration object. Use this alias to refer to this LdapServer configuration object from beectl.

Default UserObjectClass and GroupObjectClass Values

Depending on the LDAP directory type, the values of UserObjectClass and GroupObjectClass are set to one of the values specified in the following table, if those properties have not been explicitly set:

Table 14–3 Default UserObjectClass and GroupObjectClass Values

Directory/Property	UserObjectClass	GroupObjectClass
IBM_TIVOLI_DIRECTORY	inetOrgPerson	groupOfNames

Table 14–3 (Cont.) Default UserObjectClass and GroupObjectClass Values

Directory/Property	UserObjectClass	GroupObjectClass
MICROSOFT_ACTIVE_DIRECTORY	user	group
ORACLE_INTERNET_DIRECTORY	orclUserV2	orclGroup
SUN_ONE_DIRECTORY	inetOrgPerson	groupOfUniqueNames

Configuring Authentication Service to Use LDAP Server

The following steps describe how to configure the Authentication Service so that it uses your LDAP server. These steps assume that you have already enabled a synchronization profile for your LDAP server.

Notes: If you are using Active Directory as your LDAP server, refer to "[Active Directory Considerations](#)" before proceeding with these steps.

Before configuring the Authentication Service to use Active Directory, you must ensure that host names returned in an LDAP referral by Active Directory can be resolved by the Domain Name System (DNS) on the host on which you installed Oracle Beehive.

If you installed Oracle Beehive Integration for Zimbra in a separate Oracle home other than Oracle Beehive, then after configuring the Authentication Service to use your LDAP server, run the `beectl modify_local_configuration_files` command on both your Oracle Beehive Integration for Zimbra and Oracle Beehive homes.

1. Modify the `AuthStoreType` property of the Authentication Service to `ldap` with the `beectl modify_property` command:

```
beectl list_components --type AuthenticationService
```

```
-----
| Component Type          | Component Identifier |
-----
| AuthenticationService | _AuthenticationService |
-----
```

```
beectl modify_property --component _AuthenticationService
                        --name AuthStoreType --value ldap
```

```
beectl activate_configuration
```

```
beectl modify_local_configuration_files
```

Note: The `beectl modify_local_configuration_files` command will ask you to run this command on all your other instances. **Do not run this command on all your other instances at this time.** For each instance, make your desired changes to the `AuthStoreType` property and run `beectl activate_configuration` before running the `beectl modify_local_configuration_files` command.

2. To test the Authentication Service, log in with any user:

```
beectl login
--authuser newuser
--authpassword <Password of newuser, obfuscated. To use non-obfuscated
passwords, run beectl in shell mode>
User newuser is successfully authenticated and logged in.
```

To test connectivity with the LDAP server use either the commands `ldapbind` or `ldapsearch`. Refer to the documentation of your LDAP server for more information about these commands.

Configuring Digest Authentication

Digest authentication is an authentication method that involves using some known secrets (or passwords) from both the client and server to calculate a hash value. This hash value is transmitted instead of the actual secret (or password). One of the major benefits of digest authentication is that the password is not exposed while being transmitted.

To use digest authentication with a particular LDAP directory, the directory must be able to do one of the following:

- Store the user password in clear text or reversible encrypted form
- Store an A1 hash value of the password. An A1 hash value is an intermediate value used for the calculation of the authentication methods HTTP digest and SASL digest-MD5. The A1 hash value is created from a user's password, principal name (userid) and realm.

Configuring digest authentication, using an LDAP directory as the authentication repository, involves the following steps:

- [Step A: Determine Digest Mechanism Depending on LDAP Directory](#)
- [Step B: Configure Oracle Beehive](#)

Step A: Determine Digest Mechanism Depending on LDAP Directory

Determine the digest mechanism Oracle Beehive will use by referring to the following table. The digest mechanism used depends on the availability of a clear-text user password, reversible encrypted password, or a secure A1 hash value of the password. You may have to configure your LDAP directory to use certain digest mechanisms.

In the following table, each cell in the **Digest Mechanism** column specifies a digest mechanism and a list of password formats; the specified digest mechanism requires that your LDAP server is configured to store only **one** of these password formats.

Table 14–4 Supported Digest Authentication Mechanisms

Digest Mechanism	Active Directory	IBM Tivoli Directory	Oracle Internet Directory	Sun One Directory
SASL CRAM-MD5 <ul style="list-style-type: none"> ▪ Plain text ▪ Reversible encryption 	Not supported; Active directory does not allow any passwords to be read; the password attribute is write-only	Supported	Supported	Supported
SASL Digest-MD5 <ul style="list-style-type: none"> ▪ Plain text ▪ Reversible encryption ▪ A1 hash value 	Supported; extend the schema to store the A1 hash value	Supported	Supported	Supported; extend the schema to store the A1 hash value
HTTP Digest <ul style="list-style-type: none"> ▪ Plain text ▪ Reversible encryption ▪ A1 hash value 	Supported; extend the schema to store the A1 hash value	Supported	Supported	Supported; extend the schema to store the A1 hash value
SyncML v1.0 Digest <ul style="list-style-type: none"> ▪ Plain text ▪ Reversible encryption 	Not supported; Active directory does not allow any passwords to be read; the password attribute is write-only	Supported	Supported	Supported
Sync ML v1.1 Digest <ul style="list-style-type: none"> ▪ Plain text ▪ Reversible encryption 	Not supported; Active directory does not allow any passwords to be read; the password attribute is write-only	Supported	Supported	Supported

Step B: Configure Oracle Beehive

These steps assume that you have already configured Oracle Beehive to authenticate with an LDAP directory.

1. Set the properties `UseSecureHash` and `AuthenticationRealm` in the `AuthenticationService` component:
 - `UseSecureHash`: Set this property to true to use the A1 hash value. Set this property to false to use the plain or reversible encrypted password. This depends on which digest authentication method you are going to use. Refer to [Table 14–4, "Supported Digest Authentication Mechanisms"](#).
 - `AuthenticationRealm`: Default authentication realm. This value is returned to clients when digest authentication is initiated. For example, HTTP and SASL digest authentication requires that the realm value to be sent with the authentication challenge.

The following `beectl` commands set `UseSecureHash` to true and `AuthenticationRealm` to `myrealm@example.com`:

```
beectl modify_property
  --component _AuthenticationService
  --name UseSecureHash
  --value true
```

```
beectl modify_property
  --component _AuthenticationService
  --name AuthenticationRealm
```

```
--value myrealm@example.com
```

2. Set the DigestAuthenticationAttribute property in the LdapServer configuration object. Refer to ["Retrieving Information About the LDAP Server"](#) for more information about this object.

The DigestAuthentication property specifies which attributes from the user object (in the LDAP directory) are required for the digest authentication. This property can have multiple values. The format of the property value is *<Mechanism Type>: <Attribute Name>*. The following table lists the possible values for *<Mechanism Type>* and the type of attribute with which it is associated:

Table 14–5 Valid Digest Mechanism Type Values

<i><Mechanism Type></i> Value	Type of Attribute to Specify
DEFAULT	Attribute name for reversible encrypted or plain password
SASL.DIGEST_MD5	Attribute name for SASL digest authentication
HTTP.DIGEST	Attribute name for HTTP digest authentication

For example, suppose you are using Oracle Internet Directory as your LDAP server. Oracle Internet Directory stores the user password in reversible encrypted format in the attribute `orclrevpwd`. If you have also specified that Oracle Internet Directory stores the A1 hash value in the attribute `authpassword;beehive`, set `DigestAuthenticationAttribute` as follows:

```
beectl modify_property
--component <ID of LdapServer object>
--name DigestAuthenticationAttribute
--value DEFAULT:orclrevpwd
--value SASL.DIGEST_MD5:authpassword;beehive
--value HTTP.DIGEST:authpassword;beehive
```

Note: To compute and set the A1 hash value, refer to the documentation of your LDAP server. You may also use a third-party tool to create this value or create this value yourself by using the information in RFC 2617, HTTP Authentication: Basic and Digest Access Authentication.

3. Activate the configuration and commit changes.

```
beectl activate_configuration
beectl modify_local_configuration_files
```

Changing LDAP Administrator's Password

To change the administrator's password of an LDAP server synchronized with Oracle Beehive, follow these steps:

1. Stop Oracle Beehive with the `beectl stop --all` command.
2. Change the password of the LDAP administrator in your LDAP server.
3. Start Oracle Beehive with the `beectl start` command.
4. Change the password for the LDAP administrator in Oracle Beehive with the following command (obfuscate the password with the `beectl obfuscate` command.)

```
beectl modify_secure_property
--component <LdapServer of the profile>
--name LdapServerPassword
--value <Password of LDAP administrator, obfuscated>
--obfuscated
```

5. Run the `beectl modify_local_configuration_files` command.

Oracle Internet Directory Considerations

This section covers the following topics:

- [Synchronizing with Directory Replication Group](#)
- [Migrating Oracle Internet Directory from One Server to Another](#)
- [Troubleshooting Synchronization between Oracle Beehive and Oracle Internet Directory](#)

Synchronizing with Directory Replication Group

A directory replication group (DRG) consists of the directory servers that participate in the replication of a given naming context. If you have synchronized Oracle Beehive with an Oracle Internet Directory server that belongs to a multimaster DRG, then ensure that the attribute `orclDIPRepository` is set to `true`.

This ensures changes made to any server in the multimaster DRG are synchronized with Oracle Beehive.

For more information about directory replication groups, refer to Chapter 29, "Oracle Internet Directory Replication Concepts" in *Oracle Internet Directory Replication Concepts*.

Migrating Oracle Internet Directory from One Server to Another

If you migrate an Oracle Internet Directory server (that is synchronized with Oracle Beehive) to another Oracle Internet Directory server, modify the `LdapServer` property in the `_CURRENT_SITE` component with the name of the new Oracle Internet Directory server:

```
beectl list_properties --component _CURRENT_SITE
```

...

EventListenerDatabase		
SearchDatabase		
BusinessDatabase		
VirusScanEngineCluster		
SiteId	17378	
LdapServer	OLD_OID_server_example.com	
LanguagePack	byte array of size 656902	
ClusteringEnabled	true	
DiagnosabilityProperties	1f9fcf0e-7b46-427c-a6ef-636bcb88f89	
DebugProperties	19804e92-9028-49b3-af36-be4ac4abb4f5	
BtiGlobalConfiguration		
Name	R1	

```
beectl modify_property --component _CURRENT_SITE
--name LdapServer
--value <new Oracle Internet Directory server>
```

```
beectl modify_change_number
  --profile <name of your LDAP profile>
  --number <Changelog number of your new Oracle Internet Directory server>
beectl activate_configuration
beectl modify_local_configuration_files
```

Troubleshooting Synchronization between Oracle Beehive and Oracle Internet Directory

- Check the files `oidldapd.log` and `oidrepld.log` from Oracle Internet Directory.
- To retrieve change log information, the Oracle Directory Integration Server Control tool (`odisrv`) must be up and running. Refer to Chapter 2, "odisrv" in *Oracle Identity Management User Reference*.
- Make sure that the `orclDiprepository` parameter is set to true. Refer to Chapter 9, "Oracle Identity Management Attribute Reference" in *Oracle Identity Management User Reference*.
- Make sure that the value of `changed` from the `bee_data.uds_sync_profile` table is updated with the `chg_no` value from `ods.ods_chg_log` (which is a table from the Oracle Internet Directory schema). For more information about this table, see the section "LDAP-Based Replication" in Chapter 29, "Oracle Internet Directory Replication Concepts" in *Oracle Internet Directory Administrator's Guide*.

Note: Take the `chg_no` value from Oracle Internet Directory used with the cloned system and not the information from the LDAP server used by the production system.

- In Oracle Internet Directory, ensure that the attributes `krbaPrincipalName` and `orclUserApplnProvStatus` exist, otherwise create them. The `bulkload` bulk management tool might fail if these aren't defined.

For more information about Oracle Internet Directory attributes, refer to Chapter 9, "Oracle Identity Management Attribute Reference" in *Oracle Identity Management User Reference*.

For more information about `orclUserApplnProvStatus`, refer to the section "Provisioning Status in Oracle Internet Directory" in Chapter 12, "Oracle Directory Integration Platform Service Concepts" in *Oracle Identity Management Integration Guide*.

For more information about `bulkload`, refer to the section "bulkload" in Chapter 9, "Using Bulk Tools" in *Oracle Internet Directory Administrator's Guide*.

- For more information about troubleshooting Oracle Internet Database, refer to Appendix J, "Troubleshooting Oracle Internet Directory" in *Oracle Internet Directory Administrator's Guide*.

Active Directory Considerations

During authentication, the Oracle Beehive Authentication Service may request Active Directory to perform an operation, such as a search. When Active Directory has referrals enabled, instead of the search results, Active Directory may respond with an LDAP referral. This referral may point to an Active Directory instance on another host.

Ensure that the host names returned in the referral can be resolved by the Domain Name System (DNS) on the host on which you installed Oracle Beehive.

For more information about LDAP referrals, refer to the Active Directory documentation.

Troubleshooting General LDAP Synchronization Issues

Call the command `beectl validate_directory_entry` to reconcile any LDAP directory entries that are not synchronized with Oracle Beehive.

Configuring SSL

This module describes various ways to configure Oracle Beehive with SSL. It covers the following topics:

- [SSL Checklist](#)
- [Configuring SSL with Oracle Beehive](#)
- [Configuring SSL with Oracle Beehive DMZ Instances](#)
- [Procedures Related to Configuring SSL](#)

Note: Refer to "[Configuring Oracle Beehive Administration Console for SSL Access](#)" to configure SSL for Oracle Beehive Administration Console.

SSL Checklist

After following the steps described in this module, ensure the following for all your application tiers:

- A properly configured Oracle wallet resides in `<Oracle home>/Apache/Apache/conf/ssl.wlt/default` for each application tier.
- For each Oracle Beehive instance, the property `WalletDir` is set to the properly configured Oracle wallet. In addition, the property `WalletDir` refer to the same location for each application tier.
- Each Oracle Beehive instance's wallet contains a valid certificate.
- The file `<Oracle home of DMZ instance>/beehive/conf/bti.properties` is configured properly for each Oracle Beehive DMZ instance.
- The file `<Oracle home>/opmn/conf/opmn.xml` is configured properly for each application tier.

Configuring SSL with Oracle Beehive

This section covers the following procedures:

- [Configuring SSL with Test Certificates for Oracle Beehive](#)
- [Configuring SSL with Real Certificates During Installation of Oracle Beehive](#)
- [Configuring SSL with Real Certificates After Installation of Oracle Beehive](#)

Configuring SSL with Test Certificates for Oracle Beehive

The following steps describe how to configure SSL with test certificates during or after the installation of one or more Oracle Beehive instances:

1. Enable secure ONS notification for your database by following the steps described in ["Configuring Oracle Wallet for Oracle Database and Oracle RAC"](#).
2. Install your first Oracle Beehive instance, if you have not already done so.
3. By default, an Oracle wallet with test certificates for OPMN is created in Oracle Beehive. This Oracle wallet is located in `<Oracle Beehive home>/opmn/conf/ssl.wlt/default`.

Copy the contents of `<Oracle Beehive home>/opmn/conf/ssl.wlt/default` to the `<Database home>/opmn/conf/ssl.wlt/default` directory. This will overwrite the Oracle wallet files in this directory.

If you are using Oracle RAC, copy the contents of `<Oracle Beehive home>/opmn/conf/ssl.wlt/default` to the `<Database home>/opmn/conf/ssl.wlt/default` directory on each Oracle RAC node.

4. Configure TLS on your first Oracle Beehive instance. Refer to ["Configuring TLS with Oracle Wallet"](#).
5. Perform the post-install steps for configuring Oracle RAC except step 7 (Register for ONS Notification). Refer to ["Post-Install Steps"](#) in ["Configuring and Installing Oracle Beehive for Oracle RAC"](#)
6. Configure the virtual server of your Oracle Beehive instance with a load balancer. Refer to ["Configuring High Availability Environment with Load Balancer"](#) in ["Installing Oracle Beehive in High Availability Environment"](#).
7. If you have more than one Oracle Beehive instance, configure TLS on all your other Oracle Beehive instances. Refer to ["Configuring TLS on Multiple Instances"](#) in ["Configuring TLS with Oracle Wallet"](#).
8. Enable ORMIS on all your Oracle Beehive instances. Refer to ["Enabling ORMIS with Password-Protected Oracle Wallet"](#) in ["Configuring TLS with Oracle Wallet"](#).
9. Enable AJPS on all your Oracle Beehive instances. Refer to ["Enabling AJPS"](#).

Note: After configuring SSL with test (self-signed) certificates for an Oracle Beehive environment with multiple instances, you may receive an alert message similar to the following:

You have received an invalid certificate.... Your certificate contains the same serial number as another certificate issued by the certificate authority. Please get a new certificate containing a unique serial number.

In this scenario, create a self-signed certificate for each Oracle Beehive instance with a unique serial number. If you are using OpenSSL to create self-signed certificates, use the `-set_serial` option:

```
openssl x509 -req -in certreq.csr -CA cacert.crt -CAkey cakey.pem
  -CAcreateserial -set_serial 01 -days 3650 > server.crt
```

For more information about creating self-signed certificates with OpenSSL (and then importing them into Oracle Wallet), refer to ["Creating Self-Signed Certificate and Importing it into Wallet"](#).

Configuring SSL with Real Certificates During Installation of Oracle Beehive

The following steps describe how to configure SSL with real certificates during the installation of one or more Oracle Beehive instances:

1. Enable secure ONS notification on your database by following the steps described in ["Configuring Oracle Wallet for Oracle Database and Oracle RAC"](#).
2. Remove all test certificates using Oracle Wallet Manager from the wallet you created for Oracle Database in Step 1, if any. The order of removal should be (1) user certificate, (2) certificate request, and (3) trusted certificate.
3. For the wallet of Oracle Database you created in Step 1, create a self-signed server certificate for each Oracle RAC node using a root certificate (from a certificate authority). Import these self-signed server certificates as well as the root certificate to the wallet for Oracle Database. Refer to ["Creating Self-Signed Certificate and Importing it into Wallet"](#).
4. Install your first Oracle Beehive instance.
5. Configure TLS on your first Oracle Beehive instance. Refer to ["Configuring TLS with Oracle Wallet"](#).
6. Remove the test certificates using Oracle Wallet Manager from the wallets in Oracle Beehive. The order of removal should be (1) user certificate, (2) certificate request, and (3) trusted certificate. These wallets should be located in `<Oracle Beehive home>/opmn/conf/ssl.wlt/default` and `<Oracle Beehive home>/Apache/Apache/conf/ssl.wlt/default`.
7. For the wallet located in `<Oracle Beehive home>/opmn/conf/ssl.wlt/default`, create a self-signed server certificate for the Oracle Beehive server using a root certificate (from a certificate authority). Import this self-signed server certificate as well as the root certificate to this wallet. Refer to ["Creating Self-Signed Certificate and Importing it into Wallet"](#).
Repeat this step for the wallet located in `<Oracle Beehive home>/Apache/Apache/conf/ssl.wlt/default`.
8. Perform the post-install steps for configuring Oracle RAC except Step 7 (Register for ONS Notification).
9. Configure the virtual server of each Oracle Beehive instances with a load balancer. Refer to ["Configuring High Availability Environment with Load Balancer"](#) in ["Installing Oracle Beehive in High Availability Environment"](#).
10. Install an additional Oracle Beehive instance (software only install). In the following steps, this instance will be referred to as the second instance.
11. Replace `orapki` and Oracle Wallet Manager (`owm`) binaries of the second instance with those from the first instance. Create new wallets located in `<Oracle Beehive new instance home>/opmn/conf/ssl.wlt/default` and `<Oracle Beehive new instance home>/Apache/Apache/conf/ssl.wlt/default`. Refer to ["Configuring TLS with Oracle Wallet"](#).
12. Remove test certificates using Oracle Wallet Manager from the wallets in `<Oracle Beehive new instance home>/opmn/conf/ssl.wlt/default` and `<Oracle Beehive new instance home>/Apache/Apache/conf/ssl.wlt/default`, if any. The order of removal should be (1) user certificate, (2) certificate request, and (3) trusted certificate.
13. Repeat Step 8 for the second instance.

14. Run the Config Wizard for the second instance and complete the configuration.
15. Configure TLS on all Oracle Beehive instances.
16. If you want to install another Oracle Beehive instance, repeat Steps 11 to 15.
17. Enable ORMIS on all Oracle Beehive instances. Refer to ["Enabling ORMIS with Password-Protected Oracle Wallet"](#) in ["Configuring TLS with Oracle Wallet"](#)
18. Enable AJPS on all Oracle Beehive instances. Refer to ["Enabling AJPS"](#).

Configuring SSL with Real Certificates After Installation of Oracle Beehive

The following steps describe how to configure SSL with real certificates after the installation of one or more Oracle Beehive instances:

1. Enable secure ONS notification on your database by following the steps described in ["Configuring Oracle Wallet for Oracle Database and Oracle RAC"](#).
2. Remove all test certificates using Oracle Wallet Manager from the wallet you created for Oracle Database in Step 1, if any. The order of removal should be (1) user certificate, (2) certificate request, and (3) trusted certificate.
3. For the wallet of Oracle Database you created in Step 1, create a self-signed server certificate for each Oracle RAC node using a root certificate (from a certificate authority). Import these self-signed server certificates as well as the root certificate to the wallet for Oracle Database. Refer to ["Creating Self-Signed Certificate and Importing it into Wallet"](#).
4. Choose one of your Oracle Beehive instances on which to perform Steps 4 to 7 (you will repeat these steps on your other instances later). Configure TLS on the Oracle Beehive instance. Refer to ["Configuring TLS with Oracle Wallet"](#).
5. Remove the test certificates from the wallets of the Oracle Beehive instance. The order of removal should be (1) user certificate, (2) certificate request, and (3) trusted certificate. These wallets should be located in `<Oracle Beehive home>/opmn/conf/ssl.wlt/default` and `<Oracle Beehive home>/Apache/Apache/conf/ssl.wlt/default`.
6. For the wallet located in `<Oracle Beehive home>/opmn/conf/ssl.wlt/default`, create a self-signed server certificate for Oracle Beehive using a root certificate (from a certificate authority). Import this self-signed server certificate as well as the root certificate to this wallet. Refer to ["Creating Self-Signed Certificate and Importing it into Wallet"](#).

Repeat this step for the wallet located in `<Oracle Beehive home>/Apache/Apache/conf/ssl.wlt/default`.
7. Register for ONS Notification. By default, Oracle Beehive is configured to receive secure ONS notifications. If SSL ONS notification is disabled on the Oracle Beehive instance, enable it by changing the value of `NotificationServerSslEnabled` property of `OpmnCluster` component to true. Refer to [Step 8, "Register for ONS Notification"](#) in ["Configuring and Installing Oracle Beehive for Oracle RAC"](#).
8. If you have multiple Oracle Beehive instances, repeat Steps 4 to 7 for each of your instances.
9. Enable ORMIS on all Oracle Beehive instances. Refer to ["Enabling ORMIS with Password-Protected Oracle Wallet"](#) in ["Configuring TLS with Oracle Wallet"](#)
10. Enable AJPS on all Oracle Beehive instances. Refer to ["Enabling AJPS"](#).

Configuring SSL with Oracle Beehive DMZ Instances

This section covers the following procedures:

- [Configuring SSL with Test Certificates After Installation of DMZ Instances](#)
- [Configuring SSL with Real Certificates After Installation of DMZ Instances](#)

Configuring SSL with Test Certificates After Installation of DMZ Instances

The following steps describe how to configure SSL with test certificates during the installation of one or more Oracle Beehive instances:

1. Install your DMZ instance.
2. Configure Oracle Wallet for the DMZ instance. For more information, refer to "[Step A: Configuring Oracle Wallet with Oracle Beehive DMZ Instances](#)" in "[Configuring Oracle Beehive Demilitarized Zone Instances](#)". This step involves creating an Oracle Wallet for your DMZ instance and editing the file `<Oracle home of DMZ instance>/opmn/conf/opmn.xml` so that it refers to the new Oracle Wallet.
3. Follow the steps described in "[Step B: Configuring Oracle Beehive DMZ Instances](#)" in "[Configuring Oracle Beehive Demilitarized Zone Instances](#)".
4. Configure the virtual server of your Oracle Beehive DMZ instances with a load balancer. For more information, refer to "[Configuring High Availability Environment with DMZ Instances and Load Balancer](#)" in "[Installing Oracle Beehive in High Availability Environment](#)".

Configuring SSL with Real Certificates After Installation of DMZ Instances

The following steps describe how to configure SSL with real certificates after the installation of one or more Oracle Beehive DMZ instances:

1. Install your DMZ instance.
2. Configure Oracle Wallet for the DMZ instance. For more information, refer to "[Step A: Configuring Oracle Wallet with Oracle Beehive DMZ Instances](#)" in "[Configuring Oracle Beehive Demilitarized Zone Instances](#)". This step involves creating an Oracle Wallet for your DMZ instance and editing the file `<Oracle home of DMZ instance>/opmn/conf/opmn.xml` so that it refers to the new Oracle Wallet.
3. For the wallet located in `<Oracle Beehive DMZ home>/opmn/conf/ssl.wlt/default`, create a self-signed server certificate for the Oracle Beehive DMZ instance using a root certificate (from a certificate authority). Import this self-signed server certificate as well as the root certificate to this wallet. For more information, refer to "[Creating Self-Signed Certificate and Importing it into Wallet](#)".

Repeat this step for the wallet located in `<Oracle Beehive DMZ home>/Apache/Apache/conf/ssl.wlt/default`.
4. Follow the steps described in "[Step B: Configuring Oracle Beehive DMZ Instances](#)" in "[Configuring Oracle Beehive Demilitarized Zone Instances](#)".
5. Configure the virtual server of your Oracle Beehive DMZ instances with a load balancer. For more information, refer to "[Configuring High Availability Environment with DMZ Instances and Load Balancer](#)" in "[Installing Oracle Beehive in High Availability Environment](#)".

Procedures Related to Configuring SSL

This section covers the following procedures related to configuring SSL:

- [Configuring Oracle Wallet for Oracle Database and Oracle RAC](#)
- [Creating Self-Signed Certificate and Importing it into Wallet](#)

Configuring Oracle Wallet for Oracle Database and Oracle RAC

These steps create an Oracle Wallet for OPMN for Oracle Database. If you are using Oracle RAC, these steps configure Oracle Wallet for Oracle Cluster Ready Services (CRS) for each Oracle RAC node.

In the following steps, *<Database home>* refers to the location of Oracle Database, and *<CRS home>* refers to the location of CRS of your Oracle RAC node.

1. Create the directory *<Database home>/opmn/conf/ssl.wlt/default*.
2. Create the wallet with the `orapki` tool:

```
<Database home>/bin/orapki wallet create
-wallet <Database home>/opmn/conf/ssl.wlt/default
-auto_login -pwd welcome
```

This command will create a wallet in the default location, *<Database home>/opmn/conf/ssl.wlt/default*.

3. Add the following line to the file *<Database home>/opmn/conf/ons.config*:

```
walletfile=<Database home>/opmn/conf/ssl.wlt/default
```

4. If you are using Oracle RAC, then for each Oracle RAC node, repeat Steps 1, 2, and 3, and then add the following line to *<CRS home>/opmn/conf/ons.config*:

```
walletfile=<Database home>/opmn/conf/ssl.wlt/default
```

Creating Self-Signed Certificate and Importing it into Wallet

The following steps create a self-signed server certificate and imports it into an Oracle Wallet. You may also create a certificate signed by a certificate authority (CA) and import that into an Oracle Wallet. Refer to "[Creating CA-Signed Certificate and Importing it into Wallet](#)" for more information.

You will be performing these steps for the wallet you created in the following procedures:

- "[Configuring Oracle Wallet for Oracle Database and Oracle RAC](#)" (which creates a wallet for OPMN for Oracle Database)
 - "[Configuring TLS with Oracle Wallet](#)" (which creates a wallet for Oracle Beehive)
 - "[Step A: Configuring Oracle Wallet with Oracle Beehive DMZ Instances](#)" (which creates a wallet for an Oracle Beehive DMZ instance)
1. Create your own certificate authority. This step uses OpenSSL. For more information about OpenSSL, refer to <http://www.openssl.org/>.

On Linux and other UNIX-based operating systems, the command `openssl` is typically located in `/usr/bin`.

```
openssl req -new -x509 -keyout cakey.pem -out cacert.crt -days 3650
```

This command generates two files named `cakey.pem` and `cacert.crt`.

2. Create and export a certificate request with Oracle Wallet Manager:
 - a. Run Oracle Wallet manager, `<Oracle Beehive home>/bin/owm`. (Use `<Database home>/bin/owm` instead if you have not installed any Oracle Beehive instances.)
 - b. Open the wallet (to which you want to add the certificate).
 - c. Create a certificate request. Click the **Operations** tab. Click **Add Certificate Request**. Fill out the form. The **Common Name** should be the name of the server for which you are creating the certificate (such as the name of the Oracle RAC node). Click **OK**.
 - d. Save the wallet.
 - e. Click the **Operation** tab. Click **Export Certificate Request**. Enter the path and file name of the certificate request. These steps assume that the name of this file is `certreq.csr`. (Keep Oracle Wallet Manager open; you will use it in Step 4.)
3. From a command prompt, generate a server certificate with the following command:


```
openssl x509 -req -in certreq.csr -CA cacert.crt -CAkey cakey.pem
  -CAcreateserial -days 3650 > server.crt
```

This command generates two files, `cacert.crt` and `server.crt` (which is the server certificate).
4. In Oracle Wallet Manager, click the **Operations** tab. Click **Import Trusted Certificate**. Select the file `cacert.crt`. Click **OK**.
5. Click **Import User Certificate**. Select the file `server.crt`. Click **OK**.
6. Repeat Steps 2 to 5 (except Step 1; you can use the same `cakey.pem` and `cacert.crt` files for other servers) for each server for which you want to create a certificate. (In particular, you would repeat these steps for each Oracle RAC node.)

Using Oracle Wallet to Create Self-Signed Certificate

Alternatively, you may use Oracle Wallet to create a self-signed certificate.

Add a self-signed certificate to the wallet with the following command:

```
orapki wallet add
  -wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
  -dn CN=user
  -keysize 2048
  -self_signed
  -validity 3650
```

`CN=user` is the distinguished name of an arbitrary user who will be the certificate owner.

Creating CA-Signed Certificate and Importing it into Wallet

Alternatively, you may create a certificate signed by a certificate authority (CA), and import that into the Oracle Beehive wallet:

1. Add a certificate request to the Oracle Beehive wallet:

```
orapki wallet add
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-dn CN=user
-keysize 2048
-validity 3650
```

The directory `<Oracle home>/Apache/Apache/conf/ssl.wlt/default/` is the Oracle Beehive default wallet directory. `CN=user` is the distinguished name of an arbitrary user who will be the certificate owner.

2. Export the certificate request to a file:

```
orapki wallet export
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-dn CN=user
-request certificate_request.txt
```

The file `certificate_request.txt` is the exported certificate request.

3. With your certificate authority (CA) and your certificate request (`certificate_request.txt`), create a signed user certificate. In addition, export the trusted certificate from your CA. These steps use the file `user_certificate.txt` as the signed user certificate and the file `trusted_certificate.txt` as the trusted certificate exported from your CA.

You may use Oracle Wallet as a CA for testing purposes by following these steps.

a. Create an auto-login wallet to act as a certificate authority. These steps assume that this wallet is stored in `/private/ca_wallet`. Create a signed certificate from the request for test purposes:

```
orapki cert create
-wallet /private/ca_wallet
-request certificate_request.txt
-cert user_certificate.txt
-validity 3650
```

The file `user_certificate.txt` is the signed user certificate.

b. Export the trusted certificate from the CA wallet:

```
orapki wallet export
-wallet /private/ca_wallet
-dn CN=ca_user
-cert trusted_certificate.txt
```

The file `trusted_certificate.txt` is the exported (test) trusted certificate from the CA wallet.

4. Add the trusted certificate from the CA to the Oracle Beehive wallet:

```
orapki wallet add
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-trusted_cert
-cert trusted_certificate.txt
```

5. Add the user certificate to the Oracle Beehive wallet:

```
orapki wallet add
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-user_cert user_certificate.txt
```

Configuring TLS with Oracle Wallet

This module describes how to configure TLS (Transport Layer Security) with Oracle Wallet.

A wallet is a password-protected container that stores authentication and signing credentials, including private keys, certificates, and trusted certificates, all of which are used by SSL for strong authentication.

Oracle Wallet provides a TLS encrypted communication channel that some services support or require, such as XMPP, FTPS, and the Workflow Service. The following steps configure Oracle Beehive to use Oracle Wallet so that clients may access Oracle Beehive with a TLS connection.

- [Step 1: Creating Auto-Login Oracle Wallet](#)
- [Step 2: Configuring Oracle Beehive Instance to Use Oracle Wallet](#)

Refer to the section "[Changing Oracle Wallet Password](#)" to change the password of your Oracle Wallet.

Refer to the section "[Configuring TLS on Multiple Instances](#)" if you have more than one Oracle Beehive instance.

Refer to the section "[Enabling ORMIS with Password-Protected Oracle Wallet](#)" if you want to enable Oracle Remote Method Invocation over SSL.

Step 1: Creating Auto-Login Oracle Wallet

1. Ensure that the environment variable `ORACLE_HOME` is set to the home directory of Oracle Beehive.
2. Run the following command:

```
<Oracle home>/bin/orapki wallet create  
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/  
-auto_login -pwd welcome
```

This command creates a wallet with auto login enabled in the directory `<Oracle home>/Apache/Apache/conf/ssl.wlt/default/` (which is the default wallet directory) with the password `welcome`.

Step 2: Configuring Oracle Beehive Instance to Use Oracle Wallet

The following steps describe how to configure your Oracle Beehive instance to use Oracle Wallet.

1. Run the following `beectl` command:

```
beectl modify_property
  --component beehive_instance_<instance>.<host name>
  --name WalletDir
  --value <Oracle home>/Apache/Apache/conf/ssl.wlt/default
```

<instance> is the instance name you specified when you installed Oracle Beehive. To retrieve the full instance name, run the command `beectl list_components --type BeehiveInstance`.

2. Activate the configuration and restart by running the following `beectl` command:

```
beectl activate_configuration
```

Notes: If the `beectl activate_configuration` command asks you to run the `beectl modify_local_configuration_files` command, run this command.

The `beectl modify_local_configuration_files` will ask you to run this command on all your other instances. **Do not run this command on all your other instances at this time.** For each instance, you must perform steps 1 and 2 before running the `beectl modify_local_configuration_files` command.

Changing Oracle Wallet Password

Refer to "Changing the Password" in Chapter 11, "Managing Wallets and Certificates" in *Oracle Application Server Administrator's Guide*. This involves using Oracle Wallet Manager. Run `<Oracle home>/bin/owm` to run Oracle Wallet Manager.

Configuring TLS on Multiple Instances

For each instance, run all the steps required to configure TLS with Oracle Wallet.

Enabling ORMIS with Password-Protected Oracle Wallet

ORMIS is ORMI over SSL or Oracle Remote Method Invocation over Secure Socket Layer. For more information about ORMIS, refer to "Using ORMI/SSL (ORMIS) in OC4J" in Chapter 6, "Using Remote Method Invocation" in *Oracle Containers for J2EE Services Guide*.

By default, Oracle Beehive is ORMIS enabled using an anonymous cipher suite.

This section covers the following topics:

- [Disable ORMIS](#)
- [Enable ORMIS with Default SSL](#)
- [Enable ORMIS with Password Protection](#)

Disable ORMIS

1. Modify the property `_CURRENT_SITE:ManagedOc4jCluster:OrmisEnabled` to `false` and activate the configuration:

```
beectl modify_property
  --component _CURRENT_SITE:ManagedOc4jCluster
```

```
--name OrmisEnabled
--value false
```

2. Activate the configuration:

```
beectl activate_configuration
```

3. Run the command `beectl modify_local_configuration_files`. This command may restart your application tier:

```
beectl modify_local_configuration_files
```

Enable ORMIS with Default SSL

1. Modify the property `_CURRENT_SITE:ManagedOc4jCluster` to true and activate the configuration:

```
beectl modify_property
--component _CURRENT_SITE:ManagedOc4jCluster
--name OrmisEnabled
--value true
```

2. Activate the configuration:

```
beectl activate_configuration
```

3. Run the command `beectl modify_local_configuration_files`. This command may restart your application tier:

```
beectl modify_local_configuration_files
```

Enable ORMIS with Password Protection

1. Create a wallet as described in this module.
2. Modify the property `_CURRENT_SITE:ManagedOc4jCluster` to true:

```
beectl modify_property
--component _CURRENT_SITE:ManagedOc4jCluster
--name OrmisEnabled
--value true
```

3. Modify the property `WalletDir` of your Oracle Beehive instance to the path of your Oracle Wallet directory you just created with the following `beectl` commands:

```
beectl list_components --type BeehiveInstance
```

```
-----
| Component type          | Component identifier          |
-----
| BeehiveInstance      | beehive_instance_example.com |
...

```

```
beectl modify_property
--component beehive_instance_example.com
--name WalletDir
--value <Your wallet directory>
```

```
Successfully stored the property for component id
09386579-b66c-41d7-96e6-88f44673ec55.
```

4. Set the wallet password:

```
beectl modify_secure_property
--component <Component ID or alias of your Oracle Beehive instance; for
```

```
example, the previous step used beehive_instance_example.com>  
--name WalletPassword
```

This command will prompt you for the wallet password.

5. Activate the configuration:

```
beectl activate_configuration
```

6. Run the command `beectl modify_local_configuration_files`. This command may restart your application tier:

```
beectl modify_local_configuration_files
```

Configuring SSL for LDAP Integration

This module describes how to configure Oracle Beehive LDAP-based authentication with certificate authority (CA) verification. It covers the following topics:

- [Requirements](#)
- [Configure SSL for LDAP Verification](#)
- [Configure Apache HTTP Server for WebDAV Folders](#)

Requirements

Complete these tasks before configuring SSL for LDAP integration:

1. Configure your LDAP server for SSL so that your directory server authenticates itself to the client. If you are using Oracle Internet Directory as your LDAP server, choose SSL Server Authentication as your SSL authentication method. For more information, refer to Section 17.4.2, "Configure Oracle Internet Directory for SSL" in Chapter 17, "Secure Sockets Layer (SSL) and the Directory" in *Oracle Internet Directory Administrator's Guide*.
2. Request a certificate from your CA for your Oracle Beehive instance. When the CA sends the signed user certificate and its associated trusted certificate, import them into a wallet configured for your Oracle Beehive instance. Enable auto login for the wallet.

To configure an Oracle Beehive instance to use a wallet, refer to "[Configuring TLS with Oracle Wallet](#)". Refer to the following sections in *Oracle Application Server Administrator's Guide* for more information about certificates, importing certificates into Oracle Wallet, and enabling auto login:

- Section 10.5, "Certificates and Oracle Wallets" in Chapter 10, "Overview of Secure Sockets Layer (SSL) in Oracle Application Server"
- Section 11.1.3, "How to Create a Complete Wallet: Process Overview" and Section 11.1.4.14, "Using Auto Login" in Chapter 11, "Managing Wallets and Certificates"

You may use Oracle Application Server Certificate Authority as your CA. For more information, refer to *Oracle Application Server Certificate Authority Administrator's Guide*.

3. Configure LDAP with Oracle Beehive as described in "[Integrating and Synchronizing LDAP with Oracle Beehive](#)".

Configure SSL for LDAP Verification

Follow these steps to specify that your LDAP server is SSL-enabled and the location of your wallet that contains your CA certificates.

1. Get the value of `LdapServer` of your LDAP directory. This example assumes that you are using Oracle Internet Directory as your directory:

```
beectl list_properties --component oidldapdirectoryprofile
```

Property name	Property value
LdapServer	oidldapdirectoryprofile_example.com
PollInterval	15
DirectoryAttributeMap	c1bd400e-8dbe-4cf1-97c5-89b725c02f7b
ProfileState	DISABLE
DefaultFlag	DEFAULT
ProfileMode	SYNC
EnterpriseMap	dd33e82e-6842-4b24-8bf7-9a7b968ac9f1
UserTypeMap	fca1999e-7b1d-4c05-9e19-b71e52ed9c25
GroupTypeMap	35571103-caaf-4d7a-8601-90e81a5be389
Alias	oidldapdirectoryprofile

2. In the `LdapServer` object, set the value of `SslEnabled` to `true`:

```
beectl modify_property
  --component oidldapdirectoryprofile_example.com
  --name SslEnabled
  --value true
```

Changes to configuration repository are not activated.
Successfully stored the property for component id
9d2cc036-01a3-4ee6-94c8-c90311624070.

3. Get the name of your Oracle Beehive instance:

```
beectl list_components --type BeehiveInstance
```

Component type	Component identifier
BeehiveInstance	beehive_instance_example.com

4. In your Oracle Beehive instance, set the value of `walletDir` to the location of the wallet that contains your CA certificates:

```
beectl modify_property
  --component beehive_instance_example.com
  --name WalletDir
  --value <Your wallet directory>
```

Successfully stored the property for component id
09386579-b66c-41d7-96e6-88f44673ec55.

5. Run the following commands to activate your changes:

```
beectl activate_configuration
beectl modify_local_configuration_files
```

6. Restart BEECORE:

```
beectl restart --component BEECORE_example.com
```

Configure Apache HTTP Server for WebDAV Folders

After you have configured Oracle Beehive authentication with CA verification, users may get the following request every time they perform an operation on a WebDAV folder: "Choose a digital certificate. The website you want to view requests identification. Please choose a certificate."

To eliminate this request, edit the file `<Oracle_home>/Apache/Apache/conf/httpd.conf` as follows:

```
DocumentRoot "Oracle_home/Apache/Apache/htdocs"

<Directory "Oracle_home/Apache/Apache/htdocs">
  <LimitExcept GET POST OPTIONS>
    deny from all
  </LimitExcept>
  Options -FollowSymLinks -Includes -Indexes
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>
```

The only change is to append `OPTIONS` after `GET POST` in the `<LimitExcept>` tag.

With this change, when the WebDAV folder issues the `OPTIONS` HTTP access method on the document root, Apache HTTP Server will return 200 status and not issue the request for a client certificate.

Enabling AJPS

This module describes how to enable (and disable) secure Apache JServ Protocol (AJPS), so that it could be used instead of HTTP for communication between Oracle HTTP Server and OC4J.

By default, AJPS is not enabled when you install Oracle Beehive.

This module covers the following topics:

- [Enabling AJPS](#)
- [Disabling AJPS](#)

Enabling AJPS

These steps involve creating wallets and certificates. Because both of these are specific to an Oracle Beehive instances, you must perform the following steps on every Oracle Beehive instance:

1. Create a keystore with an RSA private/public key pair using the `keytool` utility.

Note: A keystore stores certificates, including the certificates of all trusted parties, for use by an application. Through its keystore, an entity such as OC4J (for example) can authenticate other parties, as well as authenticate itself to other parties. (Oracle HTTP Server uses a wallet for the same purpose.)

In Java, a keystore is a `java.security.KeyStore` instance that you can create and manipulate using the `keytool` utility that is provided with the Sun Microsystems JDK. The underlying physical manifestation of this object is a file.

For more information about the `keytool` utility, refer to <http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html>.

For additional information, refer to "Using Keys and Certificates with OC4J and Oracle HTTP Server" and "Using SSL with Standalone OC4J" in Chapter 15, "SSL Communication with OC4J" in *Oracle Containers for J2EE Security Guide*.

The following example generates a keystore in a file named `mykeystore.jks`, which has a password of `123456`, using the RSA key pair generation algorithm:

```
<Oracle home>/jdk/bin/keytool -genkey -keyalg RSA  
-keystore mykeystore.jks -storepass 123456
```

In this utility:

- The `keystore` option sets the filename where the keys are stored.
- The `storepass` option sets the password for protecting the keystore. You can optionally omit this from the command line and be prompted for a password instead.

The `keytool` utility prompts you for additional information, as follows:

```
What is your first and last name?
[Unknown]: Test User
What is the name of your organizational unit?
[Unknown]: Support
What is the name of your organization?
[Unknown]: Oracle
What is the name of your City or Locality?
[Unknown]: Redwood Shores
What is the name of your State or Province?
[Unknown]: CA
What is the two-letter country code for this unit?
[Unknown]: US
Is <CN=Test User, OU=Support, O=Oracle, L=Redwood Shores, ST=CA, C=US> correct?
[no]: yes

Enter key password for <mykey>
(RETURN if same as keystore password):
```

Note: Always press RETURN for the key password. The keystore password must be the same as the key entry password.

The `mykeystore.jks` file is created in the current directory. The default alias of the key is `mykey`.

2. Export the certificate from the keystore you just created to a file with the `keytool` utility. The following example exports the certificate into a file named `cert.txt`:

```
<Oracle home>/jdk/bin/keytool -export -file cert.txt
-keystore mykeystore.jks -storepass 123456
```

Ensure you specify the same password you used to create the keystore.

3. Import the certificate file into Oracle Wallet.
 - a. If you have not already done so, create a wallet and configure it for Oracle Beehive by following the steps described in "[Configuring TLS with Oracle Wallet](#)".
 - b. Use Oracle Wallet Manager to import the certificate. Select **Menu, Operations, Import Trusted Certificate**. Save the wallet.
4. Modify the `KeystoreFile` property of your Oracle Beehive instance:

```
beectl modify_property
--component <Oracle Beehive instance identifier>
--name KeystoreFile
--value <full path name of the keystore file>
```

To retrieve the identifier of your Oracle Beehive instance, call the following command, where `example.com` is the host name of your Oracle Beehive instance:

```
beectl list_properties --component example.com
```

```
-----+-----
Property name | Property value
-----+-----
PrimaryHostName | example.com
-----+-----
Site | _CURRENT_SITE
-----+-----
AlternateHostNames |
-----+-----
BeehiveInstances | beehive_instance_instance1.example.com
-----+-----
Alias | example.com
-----+-----
```

5. Modify the KeystoreFilePassword property (the command will prompt you for the password):

```
beectl modify_secure_property
--component <Oracle Beehive instance identifier>
--name KeystoreFilePassword
--value
```

6. Modify the AjpsEnabled property of the ManagedOc4jCluster object if it is false:

```
beectl modify_property
--component _CURRENT_SITE:ManagedOc4jCluster
--name AjpsEnabled
--value true
```

7. Modify the AjpsEnabled property of the HttpServerCluster object if it is false:

```
beectl modify_property
--component _CURRENT_SITE:HttpServerCluster
--name HttpServerSslEnabled
--value true
```

8. Commit configuration changes by calling the following beectl command:

```
beectl activate_configuration
```

Note: If the beectl activate_configuration command asks you to run the beectl modify_local_configuration_files command, run this command. The command may restart your application tier.

Disabling AJPS

1. Set the AjpsEnabled property of the ManagedOc4jCluster object to false (this example also commits configuration changes):

```
beectl modify_property
--component _CURRENT_SITE:ManagedOc4jCluster
--name AjpsEnabled
--value false
--activate_configuration true
```

2. Modify local files (the following command may restart the application tier):

```
beectl modify_local_configuration_files
```

Configuring E-Mail with SSL

You may add and configure multiple sending SMTP and receiving IMAP endpoints. As a result, you may configure Oracle Beehive e-mail so that SMTP and IMAP both listen on multiple ports; in particular, you may have SMTP and IMAP ports secured with Secure Socket Layer (SSL), and other ports with SSL not enabled.

The following topics are covered in this module:

- [Securing E-Mail with SSL](#)
- [Enabling Both Non-SSL and SSL Ports](#)
- [Adding an Endpoint to SMTP](#)
- [Modifying an Endpoint in SMTP](#)
- [Modifying or Adding an Endpoint in IMAP](#)
- [Creating and Configuring an Endpoint](#)
- [Configuring E-Mail Client](#)

Notes: The steps in this module involve changing the property `AssumeSingleEndpoint` to `false` in `SMTPProperty` and `IMAPProperty`, properties that store e-mail ports and endpoints.

If `AssumeSingleEndpoint` is `true`, then it will use the port number defined in `SMTPProperty` or `IMAPProperty`.

`SMTPProperty` and `IMAPProperty` contain a property named `EndpointPropertiesList` that contains a list of endpoints. If `AssumeSingleEndpoint` is `false`, then the port number defined in `SMTPProperty` or `IMAPProperty` will be ignored. Instead, the port numbers defined in the endpoints contained in `EndpointPropertiesList` will be used.

If you use the command `beectl modify_port` to change an e-mail port, the command assumes you want to use only one e-mail port. As a result, the command sets `AssumeSingleEndpoint` to `true` and modifies the port number defined in `SMTPProperty` or `IMAPProperty`.

Securing E-Mail with SSL

To secure Oracle Beehive e-mail with SSL, follow these steps:

1. Modify the endpoint in SMTP so that it uses an SSL-enabled port. Follow the directions in "[Modifying an Endpoint in SMTP](#)".

2. Modify the endpoint in IMAP so that it uses an SSL-enabled port. Follow the directions in ["Modifying an Endpoint in SMTP"](#) except use the property IMAPProperty in place of SMTP property.

Note: You may call the command `beectl activate_` configuration after modifying the endpoints in SMTP and IMAP, instead of calling it twice (once after modifying SMTP and again after modifying IMAP.)

Enabling Both Non-SSL and SSL Ports

1. Add or modify the endpoints in SMTP so that there are two endpoints: one endpoint that has an SSL-secured port, and one endpoint that has a non-SSL port. To add an endpoint to SMTP, refer to ["Adding an Endpoint to SMTP"](#). To modify an endpoint in SMTP, refer to ["Modifying an Endpoint in SMTP"](#).
2. Add or modify the endpoints in IMAP so that there are two endpoints: one endpoint that has an SSL-secured port, and one endpoint that has a non-SSL port. To add or modify an endpoint in IMAP, refer to ["Modifying or Adding an Endpoint in IMAP"](#).

Note: You may call the command `beectl activate_` configuration after making all your modifications in SMTP and IMAP instead of calling the command every time you make a change to either SMTP or IMAP.

Modifying an Endpoint in SMTP

1. If you want to modify the endpoint so that its port is SSL-enabled, follow the directions described in ["Configuring TLS with Oracle Wallet"](#) (if you have not already done so.)
2. Set `AssumeSingleEndpoint` to `false` in `SMTPProperty` by calling the following `beectl` commands.

Note: Setting `AssumeSingleEndpoint` to `false` in `SMTPProperty` disables the non-SSL port specified in `SMTPProperty`. (In the following example, the non-SSL port number is 25.) Instead, the configuration specified in the property `EndpointPropertiesList` (of `SMTPProperty`) is used.

The property `EndpointPropertiesList` contains endpoint objects. Each endpoint specifies a port that `SMTPProperty` uses (if `AssumeSingleEndpoint` is `false`). For each endpoint object, you may also specify whether the port is SSL-enabled or not.

```
beectl list_properties --component _EmailService:SMTPProperties
```

```
-----+-----
Property name      | Property value
-----+-----
Alias              |
-----+-----
AssumeSingleEndpoint | true
```

```

-----+-----
...
beectl modify_property --component _EmailService:SMTPProperties
  --name AssumeSingleEndpoint
  --value false
Changes to configuration repository are not activated.
Successfully stored the property for component id
  bf429057-98d4-4990-928d-a90dd6466832.

```

3. Choose an endpoint to modify from EndpointProperties list. Set EndpointScheme to `MX:*:<port number>` or `MXS:*:<port number>`, where *<port number>* is any port number not being used. Use `MX` if you do not want the port to be SSL-enabled, `MXS` otherwise. The following example sets EndpointScheme to `MXS:*:2226`:

```

beectl list_properties --component _EmailService:SMTPProperties

...
| EndpointPropertiesList | 4aa9767b-5110-4392-8717-a3c57724986b , |
|                       | f477dfe5-400d-41a2-b2a5-394ebba67293   |
...
beectl list_properties --component 4aa9767b-5110-4392-8717-a3c57724986b

...
| EndpointScheme        | MXS:\*:2226                               |
...

beectl modify_property --component 4aa9767b-5110-4392-8717-a3c57724986b
  --name EndpointScheme
  --value MXS:\*:2226
Changes to configuration repository are not activated.
Successfully stored the property for component id
  4aa9767b-5110-4392-8717-a3c57724986b.

```

4. Commit configuration changes by calling the following `beectl` commands:

```
beectl activate_configuration
```

Notes: If the `beectl activate_configuration` command asks you to run the `beectl modify_local_configuration_files` command, run this command.

The `beectl modify_local_configuration_files` command will ask you to run this command on all your other instances.

Adding an Endpoint to SMTP

1. If you want to add an endpoint with an SSL-enabled port, follow the directions described in ["Configuring TLS with Oracle Wallet"](#).
2. Set `AssumeSingleEndpoint` to `false` in `SMTPProperty` by calling the following `beectl` commands.

Note: Setting `AssumeSingleEndpoint` to `false` in `SMTPProperty` disables the non-SSL port specified in `SMTPProperty`. (In the following example, the non-SSL port number is 25.) Instead, the configuration specified in the property `EndpointPropertiesList` (of `SMTPProperty`) is used.

The property `EndpointPropertiesList` contains endpoint objects. Each endpoint specifies a port that `SMTPProperty` uses (if `AssumeSingleEndpoint` is `false`). For each endpoint object, you may also specify whether the port is SSL-enabled or not.

```
beectl modify_property --component _EmailService:SMTPProperties
--name AssumeSingleEndpoint
--value false
```

```
Changes to configuration repository are not activated.
Successfully stored the property for component id
bf429057-98d4-4990-928d-a90dd6466832.
```

3. Create and configure a new endpoint object as described in "[Creating and Configuring an Endpoint](#)". You will need the object ID of this endpoint object for the next step.
4. Add the new endpoint to the `EndpointPropertiesList` of `SMTPProperties`. Specify the new endpoint's object ID in the `--component` option:

```
beectl append_value --component bf429057-98d4-4990-928d-a90dd6466832
--name EndpointPropertiesList
--value f477dfe5-400d-41a2-b2a5-394ebba67293
```

```
Changes to configuration repository are not activated.
Successfully appended the value(s) to property EndpointPropertiesList.
```

```
beectl list_properties --component bf429057-98d4-4990-928d-a90dd6466832
```

```
-----
```

Property Name	Property Value
Port	25
*AssumeSingleEndpoint	false
*EndpointPropertiesList	4aa9767b-5110-4392-8717-a3c57724986b , f477dfe5-400d-41a2-b2a5-394ebba67293

```
...
NOTE:- * indicates that property value is changed and change is not yet
activated.
```

5. Commit configuration changes by calling the following `beectl` commands:

```
beectl activate_configuration
```

Notes: If the `beectl activate_configuration` command asks you to run the `beectl modify_local_configuration_files` command, run this command.

The `beectl modify_local_configuration_files` command will ask you to run this command on all your other instances.

Modifying or Adding an Endpoint in IMAP

Perform the steps described in or "[Modifying an Endpoint in SMTP](#)" or "[Adding an Endpoint to SMTP](#)" except use the property IMAPProperty in place of SMTPProperty.

Creating and Configuring an Endpoint

1. Create a new endpoint object with the `beectl add_configuration_object` command. This command will return the endpoint object's ID. Use this ID to configure the endpoint and view its properties:

```
beectl add_configuration_object
  --type 'oracle.ocs.management.model.EmailService.EndpointProperties'
Successfully created configuration object of type
  oracle.ocs.management.model.EmailService.EndpointProperties. This object
  can be referenced by the object-id displayed below.
Changes to configuration repository are not activated.
CONFIG_OBJECT_ID=f477dfe5-400d-41a2-b2a5-394ebba67293
```

Note: Use single quotes (') instead of double quotes (") for the `--type` option. Some command line shells may interpret strings enclosed in double quotes as variables.

2. Modify the values of the new endpoint object: The following example sets the property `EndpointName` to `My New Endpoint`, and `EndpointScheme` to `MX:*:2227`, where `2227` is a random port that is not being used.

Note: To secure this endpoint with SSL, set `EndpointScheme` to `MXS:*:<port number>`. Use `MX` instead of `MXS` if you do not want it SSL-secured.

For example, if you want your endpoint to listen on port `2227` and secure it with SSL, you would set `EndpointScheme` to `MXS:*:2227`

```
beectl list_properties --component f477dfe5-400d-41a2-b2a5-394ebba67293
```

```
-----
| Property Name      | Property Value      |
|-----|-----|
| EndpointName      | default              |
| EndpointScheme    | default              |
| Alias              |                      |
|-----|-----|
```

```
beectl modify_property --component f477dfe5-400d-41a2-b2a5-394ebba67293
  --name EndpointName --value "My New Endpoint"
```

Changes to configuration repository are not activated.
Successfully stored the property for component id
f477dfe5-400d-41a2-b2a5-394ebba67293.

```
beectl modify_property --component f477dfe5-400d-41a2-b2a5-394ebba67293
  --name EndpointScheme --value MX:\*:2227
```

Changes to configuration repository are not activated.
Successfully stored the property for component id
f477dfe5-400d-41a2-b2a5-394ebba67293.

```
beectl list_properties --component f477dfe5-400d-41a2-b2a5-394ebba67293
```

Property Name	Property Value
EndpointName	My New Endpoint
EndpointScheme	MX:*:2227
Alias	

Configuring E-Mail Client

In your e-mail client, configure your Oracle Beehive e-mail account so that the SMTP and IMAP servers use SSL and the new port numbers you specified. If you are using Outlook Express, follow these steps:

1. From the menu bar, click **Tools**, then **Accounts...**
2. In the **Internet Accounts** window, select your Oracle Beehive e-mail account and click **Properties**.
3. Click the **Advanced** tab.

For **Outgoing mail (SMTP)**, enter the port number of your SSL-enabled SMTP endpoint. For example, if the EndpointScheme of your SSL-enabled SMTP endpoint is MXS:*:2226, you would enter 2226. Select the check box **This server requires a secure connection (SSL)**.

For **Incoming Mail (IMAP)**, enter the port number of your SSL-enabled IMAP endpoint. 5144. For example, if the EndpointScheme of your SSL-enabled IMAP endpoint is MXS:*:5144, you would enter 5144. Select the check box **This server requires a secure connection (SSL)**.

Click **OK**.

Configuring Oracle Secure Enterprise Search

Oracle Secure Enterprise Search 10g is a stand-alone enterprise search solution. It incorporates best-in-class indexing crawling and security capabilities to create a reliable and comprehensive search solution for any organization.

Follow these steps to configure Oracle Secure Enterprise Search 10g so that Oracle Beehive uses it as its search engine:

Step A: Configure Oracle Beehive

1. Create a special user account that has administrator rights to the content managed by Oracle Beehive. The command will prompt you for a password. Enter any valid password; you will need this password later:

```
beectl add_trusted_identity
  --is_service false
  --service_name sessearch
  --type SES
  --name BeehiveSesTrustedEntity
  --password
```

2. Configure the host and port number of your Oracle Beehive instance:

```
beectl modify_property
  --component _SesEndptService
  --name Host
  --value <Host name of your Oracle Beehive instance>
```

```
beectl modify_property
  --component _SesEndptService
  --name Port
  --value <Oracle Beehive HTTP listening port>
```

```
beectl modify_property
  --component _SesEndptService
  --name SesEndptServiceEnabled
  --value true
```

To determine the HTTP listening port, run the `beectl list_ports` command and search for the property name `HttpListenPort`.

3. Activate and commit changes:

```
beectl activate_configuration
beectl modify_local_configuration_files
```

Step B: Configure Oracle Secure Enterprise Search

1. Login to the oracle Secure Enterprise Search administrator page. The URL typically has the form `http://<Oracle SES host name>:<HTTP listening port>/search/admin`.
2. Click the **Sources** tab. From the **Source type** drop-down list, select **Federated**. Click the **Create** button.
3. Enter the following values to define the source:
 - **Source Name:** Enter any name to identity your source
 - **Web Services URL:** `http://<Oracle Beehive host name>:<Oracle Beehive HTTP listening port>/ses-endpt/OracleSearch`
 - **Remote Entity Name:** `BeehiveSesTrustedEntity`
 - **Remote Entity Password:** Enter the password for the `BeehiveSesTrustedEntity` user you created previously.
 - **Search User Attribute:** You do not have to enter a value for this field unless Oracle Secure Enterprise Search is using a different authentication attribute than Oracle Beehive.

Configuring Single Sign-On with Oracle Beehive

This module describes how to register Oracle Beehive as a partner application with Oracle Single Sign-On (OSSO), which means that you may delegate the authentication function to the single sign-on server. Oracle Beehive. For more information, refer to Chapter 4, "Configuring and Administering Partner Applications in *Oracle Application Server Single Sign-On Administrator's Guide*.

1. In the `_AuthenticationService` component, set the property `SsoType` to `osso`, then activate the configuration:

```
beectl modify_property
  --component _AuthenticationService
  --name SsoType
  --value osso
```

```
beectl activate_configuration
```

2. Run the OSSO registration tool, `<OSSO home>/sso/bin/ssoreg.sh` to register Oracle Beehive with the OSSO server:

```
ssoreg.sh
-oracle_home_path /private/oracle/appserver
-site_name example.com>
-config_mod_osso TRUE
-mod_osso_url http://example.com:7777
-remote_midtier
-config_file
/private/oracle/appserver/Apache/Apache/conf/osso/osso.example.conf
```

- **oracle_home_path:** Specify the installation directory of the Oracle product that comes with Oracle Single Sign-On (in this example, OSSO is installed in `/private/oracle/appserver/sso`).
- **site_name:** Specify the host name (including domain) of your Oracle Beehive instance.
- **config_mod_osso:** Specify `TRUE` so that a configuration file is generated.
- **mod_osso_url:** Specify the host name and HTTP listening port of your Oracle Beehive instance. To determine the HTTP listening port, run the `beectl list_ports` command.
- **remote_midtier:** You must specify this option because Oracle Beehive is installed in a different home than OSSO.
- **config_file:** The specified configuration file will be created.

-
3. Copy the configuration file you created in the previous step (`osso.example.com`) to `<Oracle Beehive home>/Apache/Oracle/conf/osso`. Rename the file to `osso.conf`.
 4. In the `_AuthenticationService` component, set the property `OssosConfigFile` to `<Oracle Beehive home>/Apache/Oracle/conf/osso/osso.conf`, activate the configuration, then commit changes:

```
beectl modify_property
  --component _AuthenticationService
  --name OssosConfigFile
  --value <Oracle Beehive home>/Apache/Oracle/conf/osso/osso.conf
```

```
beectl activate_configuration
beectl modify_local_configuration_files
```

Configuring External Oracle BPEL Process Manager with Oracle Beehive

If you have an existing instance of Oracle BPEL Process Manager, you may configure Oracle Beehive to use it instead of the one that is included with Oracle Beehive.

Note: It is recommended that you configure an external Oracle BPEL process manager immediately after installing Oracle Beehive.

It is not recommended that you switch from using the Oracle BPEL Process Manager that comes installed with Oracle Beehive to an external one in an Oracle Beehive currently in production.

Configuring an external Oracle BPEL Process Manager involves the following steps:

- [Step A: Configure Oracle Beehive](#)
- [Step B: Configure External Oracle BPEL Process Manager](#)

After performing these steps, refer to "[Deploying Custom Workflows with External Oracle BPEL Process Manager](#)" to deploy custom workflows.

Step A: Configure Oracle Beehive

1. Create an external Oracle BPEL Process Manager component to represent your Oracle BPEL Process Manager. Use the `beectl add_external_bpel_pm` command:

```
beectl add_external_bpel_pm
  --oc4j_instance_name oc4j_soa
  --oc4j_admin_password my_obfuscated_OC4J_admin_password
  --domain default
  --domain_password my_obfuscated_domain_password
  --bpel_application_name orabpel
  --bpel_admin_password my_obfuscated_BPEL_password
  --host_name example.com
  --opmn_port 6003
```

```
Successfully created a BpelCluster with following identifier.
BPEL_CLUSTER_ID=ec542d17-0a13-4974-b6a4-f61526d09215
Successfully created a ExternalBpelProcessManager with the following id.
EXT_BPEL_PROCESS_MANAGER_ID=cb5d934a-fe61-44e0-906a-bf2d0b1c80e1
Workflow service exists with the id 79af98cb-81d1-4318-9f22-84d36e268e86
Changes to configuration repository are not activated.
```

This command will create a BpelCluster component. Note the IDs of this component for the next step.

The following table describes the properties of this command:

Table 22–1 beectl add_external_bpel_pm Properties

Property	Description
oc4j_instance_name	Name of the OC4J instance in which your Oracle BPEL Process Manager is deployed. Typically, this is <code>oc4j_soa</code> .
oc4j_admin_password	Administrator's password of OC4J in which your Oracle BPEL Process Manager is deployed. Use <code>beectl obfuscate</code> to obfuscate the password.
domain	If you have not created any additional domains in your Oracle BPEL Process Manager, typically, the name of the default domain is <code>default</code> . Refer to Chapter 19, "BPEL Process Deployment and Domain Management" in <i>Oracle BPEL Process Manager Developer's Guide</i> .
domain_password	The password to access the specified domain. By default, this is the same as the OC4J administrator's password. Use <code>beectl obfuscate</code> to obfuscate the password.
bpel_application_name	Name used by OC4J to refer to the Oracle BPEL Process Manager Web application. Typically, this is <code>orabpel</code> . Refer to the OC4J configuration files of the OC4J instance in which your Oracle BPEL process is deployed to determine this name.
bpel_admin_password	Oracle BPEL Process Manager administrator's password. Use <code>beectl obfuscate</code> to obfuscate the password.
host_name	The name of the server in which Oracle BPEL Process is running.
opmn_port	OPMN request port number of the Oracle BPEL server. This is typically 6003. If you installed Oracle Application Server SOA Suite, this port number is defined by the property <code>opmn.requestport</code> in <code><Installation directory>/bpel/utilities/and-orabpel.properties</code> .

- Set the property BpelCluster in the component `_WorkflowService` with the BpelCluster component you just created in the previous step:

```
beectl modify_property
  --component _WorkflowService
  --name BpelCluster
  --value ec542d17-0a13-4974-b6a4-f61526d09215
```

- Activate the configuration and commit changes:

```
beectl activate_configuration
beectl modify_local_configuration_files
```

Step B: Configure External Oracle BPEL Process Manager

- Create a synonym in the ORABPEL repository for the Oracle Beehive Workflow PL/SQL schema. Typically, ORABPEL and the Oracle Beehive schema are located in the same database instance. If this is the case, run the following SQL*Plus command:

```
CREATE OR REPLACE SYNONYM ORABPEL.BEE_CODE_WF_BPEL_PKG FOR
BEE_CODE.WF_BPEL_PKG
```

If ORABPEL and the Oracle Beehive schema are in different database instances, use a database link. Refer to "CREATE DATABASE LINK" in *Oracle Database SQL Language Reference* for more information.

2. Follow these steps to deploy Oracle Beehive Identify Provider.

Note: If you have configured your Oracle BPEL Process Manager for a particular LDAP directory, and you have synchronized Oracle Beehive to work with the same LDAP directory (as described in ["Integrating and Synchronizing LDAP with Oracle Beehive"](#)), do not perform the steps described here.

- a. Extract the contents of `<Oracle Beehive home>/beehive/seed/bpel/isprovider.jar` into the `<Oracle BPEL Process Manager home>/bpel/system/classes` directory. The command `jar` is the Java archive tool:

```
cd <Oracle BPEL Process Manager home>/bpel/system/classes
jar xvf <Oracle Beehive home>/beehive/seed/bpel/isprovider.jar
```

- b. Create a backup of the file `<Oracle BPEL Process Manager home>/bpel/system/config/is_config.xml`. Replace this file with `<Oracle Beehive home>/beehive/seed/bpel/is_config.xml`

After performing these steps, any identity services you have configured for your Oracle BPEL Process Manager will no longer work.

If you want Oracle Beehive Identity Provider and another identity service you have already configured to co-exist in your Oracle BPEL Process Manager, you may merge the details of your identity service configuration and those of Oracle Beehive Identity Provider in the same `is_config.xml` file. Refer to "Configuring the Identity Service" in Chapter 2, "Service Configuration" in *Oracle BPEL Process Manager Administrator's Guide*.

3. Deploy Oracle Beehive workflows. The Workflow service deploys two BPEL processes, `ParallelVoteWorkflow` and `SerialWorkflow`, when you install Oracle Beehive. These BPEL process suitcases (or archives) are located in `<Oracle Beehive home>/beehive/seed/workflows`. Deploy these processes with the Oracle BPEL Admin Console or auto-deploy mode of Oracle BPEL Process Manager. To use auto-deploy mode, copy the BPEL process jar files into `<Oracle BPEL Process Manager home>/bpel/domains/default/deploy`.

Deploying Custom Workflows with External Oracle BPEL Process Manager

To deploy a custom workflow with an external Oracle BPEL Process Manager, you must both register it in Oracle Beehive and deploy it through your Oracle BPEL Process Manager:

1. Use the `beectl add_workflow_schema` command to register your custom workflow:

```
beectl add_workflow_schema
  --file <Absolute path of the suitcase (the jar file) of your BPEL process>
```

2. Manually deploy the custom workflow suitcase into your Oracle BPEL Process Manager with the Oracle BPEL Admin Console or auto-deploy mode.

Installing Oracle Beehive Integration for Outlook

This module describes how to install and configure Oracle Beehive Integration for Outlook.

- Requirements
- Installing Oracle Beehive Integration for Outlook Using the Remote Downloader
- Installing Oracle Beehive Integration for Outlook Interactively
- Installing Oracle Beehive Integration for Outlook Silently
- Uninstalling Oracle Beehive Integration for Outlook
- Installing Oracle Beehive Integration for Outlook Using Active Directory
- Updating Oracle Beehive Integration for Outlook in the DMS Repository
- Applying the Language Pack for Oracle Beehive Integration for Outlook
- Applying Configuration Settings from the DMS
- Customizing General Installation Settings
- Downgrading your Oracle Beehive Integration for Outlook Installation
- Registry Settings
- INI File Settings
- Enforcing Upgrades
- Sample XML Provisioning File

Requirements

Microsoft Windows XP Service Pack 2 or later is required on Windows XP computers where users intend to leverage non-English locales with Oracle Beehive Integration for Outlook. Attempting to run Oracle Beehive Integration for Outlook with a non-English locale on a computer with Windows XP Service Pack 1 (or earlier) may result in the computer failing to operate normally.

Installing Oracle Beehive Integration for Outlook Using the Remote Downloader

You can install Oracle Beehive Integration for Outlook by distributing a **remote downloader** to end users. The remote downloader allows end users to download and

install the provisioned Oracle Beehive Integration for Outlook application from the Device Management Service (DMS) repository. By default, the DMS contains a pre-seeded Oracle Beehive Integration for Outlook application that is provisioned for all end users. For more information on how to provision applications for specific users or groups of users, see the "Managing the Device Management Service" module in the *Oracle Beehive Administrator's Guide*.

Retrieve the remote downloader `outlook_integration_downloader.exe` at `<Oracle-Home>/beehive/bootstrap/obio/downloader`, where `<Oracle-Home>` is the name of your Oracle Beehive directory.

Pre-Seeding a Server Name in the Remote Downloader

The remote downloader prompts the end user to provide a server name, account name, and password to initiate download and installation. If you wish to pre-seed a server name to prevent end users from having to specify a server name on their own, rename the remote downloader to `<fully qualified server name>.exe`. For example, if your server name is `faulkner.oracle.com`, your remote downloader name would be `faulkner.oracle.com.exe`.

Pre-Seeding Port Number and Secure Connection

You can pre-seed the port number that Oracle Beehive Integration for Outlook should use when connecting to the server, as well as whether a secure connection should be used, by using commas to delimit the additional information. For example, if your remote downloader name is `faulkner.oracle.com.exe` and you wish to pre-seed the port number and secure connection in the downloader, you would rename the file to `faulkner.oracle.com,5223,SSL.exe`, where 5223 is your port number and SSL is the secure connection.

Note: You can only pre-seed port number and connection type when using the direct connection method for Oracle Beehive Integration for Outlook.

Installing Oracle Beehive Integration for Outlook Interactively

You can give end users direct access to the Oracle Beehive Integration for Outlook installer. End users install the product by double-clicking the installer and following the on-screen instructions.

Retrieve the installer `outlook_integration_setup.exe` at `<Oracle-Home>/beehive/bootstrap/obio/setup`, where `<Oracle-Home>` is the name of your Oracle Beehive directory.

Installing Oracle Beehive Integration for Outlook Silently

You can install Oracle Beehive Integration for Outlook silently without users having to provide input or perform an installation themselves.

To install Oracle Beehive Integration for Outlook silently:

Execute the following command:

```
outlook_integration_setup.exe /s UI=<Installer-UI-Option>
```

`<Installer-UI-Option>` can be one of the four values provided in the table below.

Table 23–1 Installer UI Options

Value	Function
1	No user interface during install
2	Displays only a progress bar during install
3	Presents an install screen with different dialog boxes but does not require user input to run
4	Runs a fully interactive installer requiring user input

To uninstall Oracle Beehive Integration for Outlook in silent mode, use the `Msiexec.exe /x {A634A2AF-2495-4F36-B88F-0B24B84A183C} /qn` command.

Uninstalling Oracle Beehive Integration for Outlook

End users can remove Oracle Beehive Integration for Outlook from their computers by clicking **Add/Remove Programs**, selecting Oracle Beehive Integration for Outlook from the list, and clicking **Remove**.

Installing Oracle Beehive Integration for Outlook Using Active Directory

If your end users are running systems on Windows 2000, Windows XP, or Windows Vista that are locked down, they can install Oracle Beehive Integration for Outlook using Active Directory Group Policy. To perform an Active Directory installation, the domain administrator must create a package for Oracle Beehive Integration for Outlook in the Active Directory Administrative Tool and then assign the package to the organization units that contain the intended end users. The software will then be installed on end users' computers during startup.

Creating an msi installation package for Oracle Beehive Integration for Outlook

Active Directory only recognizes software packaged in the `msi` file format, which contains information about the application setup and installation.

To create an msi installation package for Oracle Beehive Integration for Outlook:

1. Extract the `outlook_integration_application.msi` file from the Oracle Beehive Integration for Outlook package using the `outlook_integration_setup.exe /export type=client <Extracted folder path>` command.
2. Extract the `prerequisite.msi` file from the Oracle Beehive Integration for Outlook package using the `outlook_integration_setup.exe /export type=system <Extracted folder path>` command.

Deploying Oracle Beehive Integration for Outlook using Active Directory Group Policy

Before proceeding, make sure you have extracted the `msi` files for Oracle Beehive Integration for Outlook to an accessible location.

To deploy Oracle Beehive Integration for Outlook using Active Directory Group Policy:

1. From the **Start** menu, select **Control Panel**, then **Administrative Tools**.
2. Click **Active Directory Users and Computers**.

3. Create an organization unit that includes all the computers on which you want to install Oracle Beehive Integration for Outlook. Alternatively, select an existing organization unit.
4. Right-click your chosen organization unit in the **Active Directory Users and Computers** tree.
5. From the **Shortcut** menu, click **Properties**.
6. In the **Properties** dialog box, click the **Group Policy** tab.
7. Click **Edit** if you are modifying an existing group policy, or **Add** if you are creating a group policy.

Note: The following steps will ensure that Oracle Beehive Integration for Outlook is installed on all computers in your chosen organization unit. It also ensures that all users of a given computer have access to Oracle Beehive Integration for Outlook.

8. Select and expand the **Computer Configuration** node.
9. Under the **Computer Configuration** node, expand the **Software Settings** folder.
10. Right-click **Software Installation** and select **New**.
11. From the **Shortcut** menu, click **Package**.
12. Enter the path to your extracted **prerequisite.msi** file.
13. Selected **Assigned** and click **OK**.
14. Repeat steps 10-13 for your extracted **outlook_integration_application.msi** file.

Note: For more information on how to obtain msi files for Oracle Beehive Integration for Outlook, see [Creating an msi installation package for Oracle Beehive Integration for Outlook](#).

15. In the **Properties** dialog box, click **OK**.
16. Exit the **Active Directory Users and Computers** console.

Verifying that Oracle Beehive Integration for Outlook is available for installation

To verify that Oracle Beehive Integration for Outlook is available to end users for installation, restart any computer in the domain and make sure that Oracle Beehive Integration for Outlook installation starts when the computer restarts.

Updating Oracle Beehive Integration for Outlook in the DMS Repository

You can update the existing version of the Oracle Beehive Integration for Outlook package in the DMS repository using the `beectl` command-line tool located at `<Oracle-Home>/beehive/bin`. If Oracle Beehive Integration for Outlook is running on an end user's machine, it will automatically prompt the end user to download and install updates from the DMS repository. End users can also manually search for updates by selecting **About Oracle Beehive Integration for Outlook** from the **Help** menu and clicking **Check for Updates**.

To update Oracle Beehive Integration for Outlook in the DMS repository:

1. Make sure the application package for the updated version of Oracle Beehive Integration for Outlook is accessible to the Oracle Beehive server.
2. Obtain your enterprise identifier by executing the following command:
3. Upload the updated version of Oracle Beehive Integration for Outlook to the DMS by executing the following command:

```
beectl list_enterprises --entity_format id
```

```
beectl upload_client_application --file <Path-to-Oracle
Beehive Integration for Outlook-Zip-File> --community
<Your-Enterprise-Identifier>
```

4. For Oracle Beehive Release 1 (1.2) only, provision end users for the uploaded version of Oracle Beehive Integration for Outlook by executing the following command:

```
beectl add_client_application_provisioning
--community <Your-Enterprise-Identifier>
--all
```

Notes: Step 4 is required only for Oracle Beehive Release 1 (1.2). It will provision all client applications in the DMS for the specified community. For information on how to provision a single application for a specific community, refer to "Managing the Device Management Service" in *Oracle Beehive Administrator's Guide*.

For Oracle Beehive Release 1 (1.3) and later, the `--community` option of the `beectl upload_client_application` command also accepts the distinguished name of the enterprise. For example:

```
beectl list_enterprises
```

```
-----+-----
Enterprise name | Identifier
-----+-----
Oracle          | enpr=Oracle
-----+-----
```

You may use `enpr=Oracle` for the enterprise identifier instead.

Applying the Language Pack for Oracle Beehive Integration for Outlook

You can localize the user interface of Oracle Beehive Integration for Outlook to match Outlook languages by applying the language pack found at the following location:

```
<Oracle-Home>/beehive/bootstrap/obio/languagepack
```

The following languages are currently supported in the language pack:

- French
- German
- Italian
- Spanish
- Brazilian Portuguese

- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

To apply the language pack for Oracle Beehive Integration for Outlook, you must upload it to the DMS repository using the `beectl` command-line tool located at `<Oracle-Home>/beehive/bin`.

To apply the language pack for Oracle Beehive Integration for Outlook:

1. Make sure the language pack for Oracle Beehive Integration for Outlook is accessible to the Oracle Beehive server.
2. Obtain your enterprise identifier by executing the following command:
3. Upload the language pack for Oracle Beehive Integration for Outlook to the DMS by executing the following command:

```
beectl list_enterprises --entity_format id
```

```
beectl upload_client_application --file <Path-to-Oracle
Beehive Integration for Outlook-Language-Pack> --community
<Your-Enterprise-Identifier>
```

4. For Oracle Beehive Release 1 (1.2) only, provision end users for the uploaded language pack by executing the following command:

```
beectl add_client_application_provisioning
--community <Your-Enterprise-Identifier>
--all
```

Notes: Step 4 is required only for Oracle Beehive Release 1 (1.2). It will provision all client applications in the DMS for the specified community. For information on how to provision a single application for a specific community, refer to "Managing the Device Management Service" in *Oracle Beehive Administrator's Guide*.

For Oracle Beehive Release 1 (1.3) and later, the `--community` option of the `beectl upload_client_application` command also accepts the distinguished name of the enterprise. For example:

```
beectl list_enterprises
```

```
-----+-----
Enterprise name | Identifier
-----+-----
Oracle          | enpr=Oracle
-----+-----
```

You may use `enpr=Oracle` for the enterprise identifier instead.

If Oracle Beehive Integration for Outlook is running on an end user's machine, it will automatically prompt the end user to download and install updates (including language packs) from the DMS repository. End users can also manually search for updates by selecting **About Oracle Beehive Integration for Outlook** from the **Help** menu and clicking **Check for Updates**.

Note: The language installed by the Oracle Beehive Integration for Outlook language pack will automatically match the Outlook language installed by the end user, provided that Oracle Beehive Integration for Outlook supports the language. If the end user has installed Outlook in an unsupported language, Oracle Beehive Integration for Outlook will default to English.

Applying Configuration Settings from the DMS

When deploying Oracle Beehive Integration for Outlook from the DMS, you can apply customized configuration settings to end users' computers.

To apply configuration settings from the DMS:

1. Create an XML provisioning file that includes all the setting parameters you want to apply. For more information, see [Customizing General Installation Settings](#).
2. Obtain your Oracle Beehive Integration for Outlook identifier by executing the `beectl list_client_applications` command and locating the identifier from the generated list.
3. Obtain your Oracle Beehive Integration for Outlook version identifier by executing the following command:

```
beectl list_client_application_versions --application
<Identifier>
```

<Identifier> is the Oracle Beehive Integration for Outlook identifier you obtained in the previous step.

4. Include the obtained Oracle Beehive Integration for Outlook version identifier in your XML provisioning file. For more information on `beectl` commands, see the module "Oracle Beehive Command-Line Utility" in the *Oracle Beehive Administrator's Reference Guide*.
5. Upload your XML provisioning file to the server by executing the following command:

```
beectl add_client_application_configuration --file
<Path-to-XML-Provisioning-File>
```

6. Provision end users for the uploaded application configuration object by executing the following command:

```
beectl add_client_application_provisioning
--community <Your-Enterprise-ID>
--configuration <Your-Application-Configuration-ID>
```

Customizing General Installation Settings

You can control general settings by creating an XML provisioning file and adding or modifying Property nodes in the `config` section of the file. Each Property node should contain two parameters: name and value.

You can modify the default public properties in your provisioning file by changing the value parameter associated with any of the `INSTALL: <X>` Property nodes.

Table 23–2 Public Properties

Property	Function	Example
USERNAME	Sets the name of the registered Oracle Beehive Integration for Outlook end user that appears in the About box.	Edith Wharton
COMPANYNAME	Sets the name of the registered company that appears in the About box.	Oracle
INSTALLDIR	Sets the location of the Oracle Beehive Integration for Outlook installation.	C:\Program Files\Oracle\myinstall
CREATE_PROFILE	Determines if an Oracle Beehive Integration for Outlook profile should be created on the user's machine during installation. This parameter can be assigned one of the following values: <ul style="list-style-type: none"> ■ 0: No profile is created. ■ 1: Force-add a profile with the specified name. ■ 2: Create a profile only if the end user is running a fresh install of Oracle Beehive Integration for Outlook. This is the default value for the parameter. ■ 3: Create a profile only if no profile with the same name exists on the end user's computer. 	Not applicable
PROFILE_NAME	Changes the name of the profile created during installation of Oracle Beehive Integration for Outlook. The default profile name is Beehive.	

The values you set for the first three parameters in the table only apply when you are installing Oracle Beehive Integration for Outlook for the first time and do not apply to product upgrades.

Downgrading your Oracle Beehive Integration for Outlook Installation

By default, Oracle Beehive Integration for Outlook will only detect updates if the server version is newer than the currently installed version. If you wish to allow end users to install downgraded versions of Oracle Beehive Integration for Outlook as updates, set the value of the `INSTALL:AllowDowngrade` Property node to `TRUE` in your provisioning file.

Registry Settings

You can modify the registry information on end-user machines by changing the values for certain `Property` nodes in your provisioning file. Modifying registry information allows you to control end user settings from the server side.

WARNING: If you modify the registry incorrectly, serious problems may occur that could require you to reinstall your operating system. Modify the registry at your own risk.

Add a key or registry value using the following coding structure:

```
<Property name="REG:ADD:{HIVE}\{keypath}:{Value Name}:{DataType}" value="{Value Data}"></Property>
```

- The value for *{HIVE}* can be either HKLM or HKCU, which are the two locations where user registry information can be stored and which correspond to the HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER hives.
- The value for *{DataType}* can be String, Binary, or Dword.
- If you are adding a key, you do not need to provide values for *{Value Name}*, *{DataType}*, or *{Value Data}*.

Delete a key or registry value using the following coding structure:

```
<Property name="REG:Delete:{HIVE}\{keypath}:{Value Name}" value=""></Property>
```

- If you are deleting a key, do not provide a value for *{Value Name}*. Additionally, the key must reside at least two levels under the user hive.

INI File Settings

You can modify Oracle Beehive Integration for Outlook ini settings by including a property with the following coding structure:

```
<Property name="FILE:Beehive.ini:{Section Name}:{Property Name}" value="{Value Data}"></Property>
```

Enforcing Upgrades

You can force end users to upgrade to a new version of Oracle Beehive Integration for Outlook that you have uploaded to the DMS. End users will not be able to use Oracle Beehive Integration for Outlook until they upgrade.

To force end users to upgrade to a new version of Oracle Beehive Integration for Outlook:

In your XML provisioning file, add a property with the following coding structure:

```
<Property name="INSTALL:ForceClientVersion:{Comparison Operation}" value="{Build-Number}"></Property>
```

{Build-Number} is the Oracle Beehive Integration for Outlook build number which will be used for the comparison operation.

Allowing a Grace Period Before Enforcing Upgrades

Optionally, you can allow end users to continue using their old version of Oracle Beehive Integration for Outlook for a specific period of time or for a fixed number of log-ins to Oracle Beehive.

To allow end users to use an older version of Oracle Beehive Integration for Outlook until a certain date:

In your XML provisioning file, add a property with the following coding structure:

```
<Property name="INSTALL:graceperioddate" value="{YYYYMMDD}"></Property>
```

{YYYYMMDD} is the final date on which the user can use Oracle Beehive Integration for Outlook without having to upgrade.

To allow end users to use an older version of Oracle Beehive Integration for Outlook for a fixed number of log-ins:

In your XML provisioning file, add a property with the following coding structure:

```
<Property name="INSTALL:graceperioditerations"
value=" {Number-of-Log-ins}"></Property>
```

{Number-of-Log-ins} is the number of times end users are allowed to log in to Oracle Beehive Integration for Outlook before they are forced to upgrade.

Sample XML Provisioning File

```
<?xml version="1.0" encoding="UTF-8"?>
<ClientAppConfig>
<Name>Oracle Beehive Integration for Outlook-test2</Name>
<Version
CollabId="6EAB:574C:capv:4672A63200D4B462E040548C17821C2E000000014C0A"/>
<Config>

<!--Install Company name-->
<Property name="INSTALL:COMPANYNAME" value="TestCompany"></Property>

<!--Install user name -->
<Property name="INSTALL:USERNAME" value="Edith Wharton"></Property>

<!--Install directory-->
<Property
name="INSTALL:INSTALLDIR"
value="C:\Program Files\Oracle\myinstall"></Property>

<!--Allow Downgrade-->
<Property name="INSTALL:AllowDowngrade" value="TRUE"></Property>

<!--Install Create Profile-->
<Property name="INSTALL:CREATE_PROFILE" value="2"></Property>

<!--Install Profile Name-->
<Property name="INSTALL:PROFILE_NAME" value="OracleBeehive"></Property>

<!--Install ForceUpgrade-->
<Property name="INSTALL:ForceClientVersion:Larger" value="20201"></Property>

<!--Install Grace Period for ForceUpgrade-->
<Property name="INSTALL:graceperioddate" value="20080215"></Property>
<Property name="INSTALL:graceperioditerations" value="3"></Property>

<!--ini File Modification-->
<Property
name="FILE:Beehive.ini:OBIO:APP-AUTOUPDATE-STARTUP"
value="FALSE"></Property>

<!--Registry Modification-->
<!--ADD Section-->
<!--ADD Key-->
<Property name="REG:ADD:HKCU\Software\1\2\3:" value=""></Property>
<!--ADD String Value (remove line breaks from the
```



```
        value of the name parameter) -->
<Property name=
  "REG:ADD:HKLM\Software\Microsoft\Exchange\Client\Extensions:
  Outlook Setup Extension1:String"
  value="4.0;Outxxx.dll;7;00000000000000;0000000;OutXXX"></Property>
<!--ADD DWORD Value (remove line breaks from the
  value of the name parameter) -->
<Property name="REG:ADD:HKLM\Software\Microsoft\Exchange\Client\Extensions:
  Outlook Setup Extension2:DWORD" value="50"></Property>
<!--ADD Binary Value (remove line breaks from the
  value of the name parameter) -->
<Property name="REG:ADD:HKLM\Software\Microsoft\Exchange\Client\Extensions:
  Outlook Setup Extension3:Binary" value="1234567890ABCDEF"></Property>

<!--Delete Section-->
<!--Delete Key-->
<Property name="REG:DELETE:HKCU\Software\1\2" value=""></Property>
<!--Delete Value (remove line breaks from the
  value of the name parameter)-->
<Property name="REG:DELETE:HKLM\Software\Microsoft\Exchange\Client\Extensions:
  Outlook Setup Extension1" value=""></Property>

</Config>

</ClientAppConfig>
```

Cloning Oracle Beehive Application Tiers and Sites

This module covers the following topics:

- [Introduction to Cloning](#)
- [Application Tier Cloning](#)
- [Site Cloning](#)
- [OC4J Instance Cloning](#)
- [Oracle Beehive Integration for Zimbra Cloning](#)

This module also covers the following topics about customizing and troubleshooting the cloning process:

- [Customizing Files or Directories in a Cloned Image](#)
- [Customizing Ports in a Cloned Instance](#)
- [Cloning Application Tiers and Sites with Ports Less Than 1024](#)
- [Oracle Inventory Location Option of Clone Commands on UNIX-Based Systems](#)
- [Cloned Application Tiers Are Not Automatically SSL or AJP Enabled](#)
- [Cloned Application Tiers and LDAP Synchronization](#)
- [Replicating LDAP Server for Cloned Instance](#)
- [Site Cloning and Multiple Instances](#)
- [References to Oracle Application Server Cloning Documentation](#)

Introduction to Cloning

Cloning is the process of copying an existing installation to a different location while preserving its configuration.

Cloning enables you to safely modify an existing Oracle Beehive instance in production, such as installing a new patch or making changes to the database. Clone your existing Oracle Beehive instance and apply your changes to the clone. Once you have verified and certified that your changes work as expected, you may safely apply those changes to your Oracle Beehive instance in production.

A cloned installation behaves the same as the source installation. For example, you can uninstall or patch the cloned instance with the Oracle Beehive Install Wizard. You can also use a cloned installation as the source for another cloning operation.

The cloning process works by copying all files from the source Oracle home to the destination Oracle home. Hence, the cloning process does not copy any files used by the source instance that are located outside the source Oracle home's directory structure. After the files are copied, a set of `beectl` commands are used to update the information in key configuration files.

Note: A wallet (if one has been configured in the source Oracle home) will be copied to the destination Oracle home. However, the clone will deliberately stop referring to the location of the wallet in the cloned application tier. You will have to manually reconfigure the wallet for the cloned application tier.

Do not overwrite the wallet in the cloned application tier.

Oracle Beehive services deployed in the source instance are also copied to the cloned instance and automatically deployed.

This module describes three cloning procedures: "[Application Tier Cloning](#)", "[Site Cloning](#)" and "[OC4J Instance Cloning](#)".

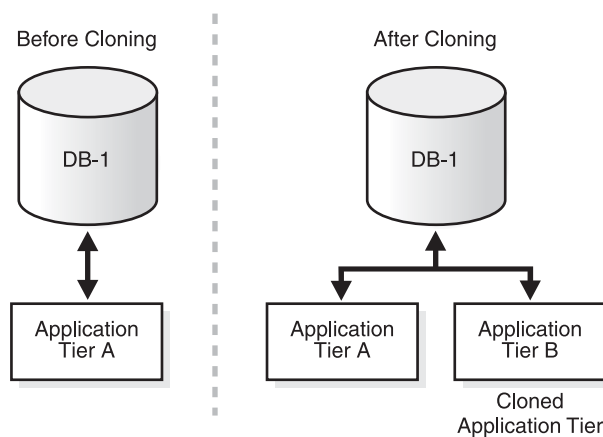
Application Tier Cloning

Application tier cloning involves the following steps:

1. **Preparing the source:** This step involves creating an image of the application tier to clone, calling the `beectl clone_preparation` command, and archiving the required files in the Oracle home in a zip file. The cloned application tier is called the *source instance* or *source image*, and the zip file is called the *clone image*.
2. **Cloning the application tier:** This step involves creating a new application tier. It involves unzipping the clone image and calling the `beectl clone_midtier` command. The new application tier is called the *cloned application tier* or *target application tier*.

The following image illustrates where a cloned application tier is located in relation to the database and the source application tier:

Figure 24–1 Application Tier Cloning



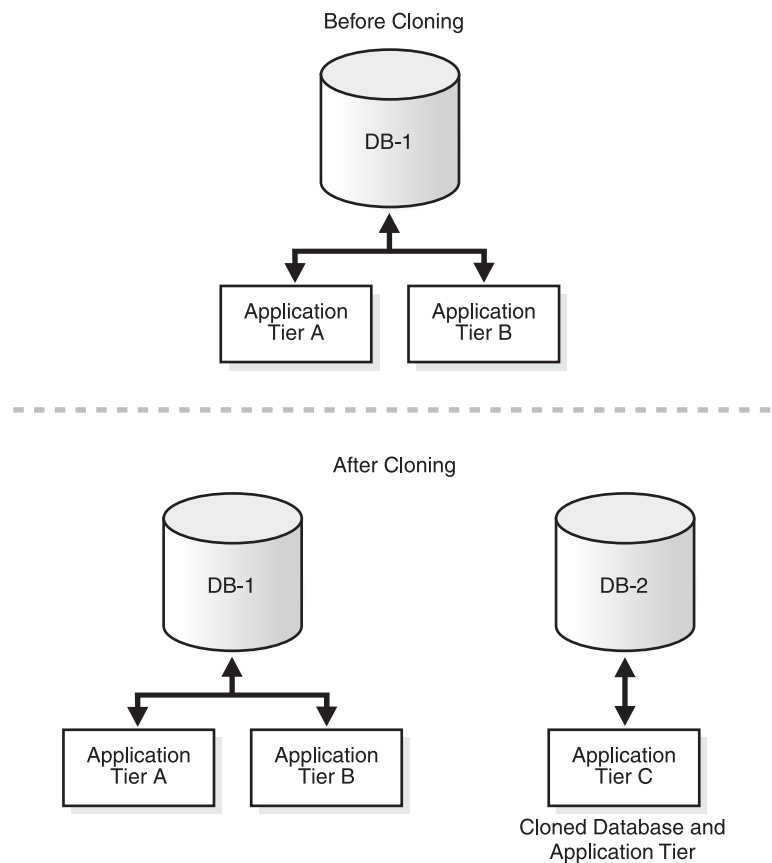
Site Cloning

Site cloning involves the following steps:

1. **Cloning the information in the database:** Oracle Beehive stores its centralized configuration information and business data in Oracle Database. This step involves cloning this data using standard database backup-recovery procedures (such as RMAN and export-import) into a new instance of Oracle Database. This ensures no overlap or sharing between the existing deployment (the cloned instance) and the new site (the new instance).
2. **Preparing the source:** This step is the same as the one described in "[Application Tier Cloning](#)".
3. **Cloning the application tier instance in the new site:** This step creates an application tier in the new site. It involves unzipping the image created during the preparing the source step, and calling the `beectl clone_site` command.

The following image illustrates where a cloned site (a cloned application tier and its database) is located in relation to the source database instance and source application tiers:

Figure 24–2 Site Cloning



Note: As [Figure 24–2, "Site Cloning"](#) illustrates, site cloning will create a single application tier in the cloned site irrespective of the number of application tiers in the source site.

To create multiple application tiers in the cloned site, follow the procedures for application tier cloning for each application tier.

Note that the cloned application tier will not be SSL enabled if the source application tier was.

OC4J Instance Cloning

OC4J instance cloning allows you to clone Oracle Beehive's managed components. When you clone an Oracle Beehive managed component, a new OC4J instance is created and all the services in the source managed component are deployed in the newly cloned OC4J instance.

Oracle Beehive Integration for Zimbra Cloning

You may clone an instance of Oracle Beehive Integration for Zimbra. Refer to "[Oracle Beehive Integration for Zimbra Cloning](#)" for more information.

Preparing Source Application Tier Instance

Follow these steps to create a source image of the application tier you want to clone:

- [Step 1: Verify Requirements](#)
- [Step 2: Clear or Activate Any Pending Configuration Changes to the Central Configuration Repository](#)
- [Step 3: Unset Environment Variables](#)
- [Step 4: Shut Down All Processes On the Application Tier](#)
- [Step 5: Call `beectl clone_preparation` Command](#)
- [Step 6: Zip Files to Create Clone Image](#)

Step 1: Verify Requirements

Ensure that the chipset and the operating system version of the source machine is the same as the destination machine. (The source machine contains the installation of Oracle Beehive you want to clone.)

Windows

You must have administrator privileges to clone an installation of Oracle Beehive.

Step 2: Clear or Activate Any Pending Configuration Changes to the Central Configuration Repository

The cloning commands update the central configuration repository (the database tables of the Oracle Beehive schema).

If you have made any changes to the central configuration repository (for example, by running the `beectl modify_property` command), activate those changes by running `beectl activate_configuration` or clear them by running `beectl clear_proposed_configuration`.

Not activating or clearing any pending configuration changes may hinder the cloning process.

Step 3: Unset Environment Variables

Unset the environment variables `ORACLE_HOME`, `ORACLE_SID`, and `LD_LIBRARY_PATH` (if they have been set or defined in your system).

Oracle application scripts invoked by `beectl` cloning commands may fail if you do not unset these environment variables.

Step 4: Shut Down All Processes On the Application Tier

Call the command `beectl stop --all` to shut down all processes running in the source Oracle home.

Note: Shutting down all processes in the application tier is *not* strictly required but it is advisable. This ensures that none of the files in the Oracle home are in use. You will later archive the Oracle home into a zip file. You may receive warnings or errors from your zip tool if some files are in use.

Step 5: Call `beectl clone_preparation` Command

The `beectl clone_preparation` command calls the Oracle Application Server `prepare_clone.pl` script, which creates local copies of several files that contain information useful for the cloning process. For example, this command creates a file in the Oracle home that contains the current host name and Oracle home path. The cloning process uses this information to search for and replace various strings in local configuration files on the target application tier.

This command also outputs a list of files (relative to the location of Oracle home) required to be zipped up to create the clone image.

The cloning commands (`beectl clone_middtier` and `beectl clone_site`) will fail if you have not called the `beectl clone_preparation` command previously:

```
beectl clone_preparation --file <fully qualified file name>
```

The following table describes the options for the `beectl clone_preparation` command:

Table 24–1 *beectl clone_preparation Options*

Option	Mandatory/ Optional	Description
--file	Mandatory	<p>The name of the text file that will be created by the <code>beectl clone_preparation</code> command. This text file will contain the names of files in the source Oracle home to be copied for cloning to the target location.</p> <p>All the files in source Oracle home need not be copied because log files, cache data, and other security files specific to the Oracle home will not be useful on the cloned Oracle home and may present a security concern.</p> <p>Note: This text file must not be located in the Oracle home because you may receive warnings or errors from some zip tools about zipping an open file.</p>

Step 6: Zip Files to Create Clone Image

First, verify that the command `<Oracle home>/beehive/bin/hasbind` is owned by the user who installed your Oracle Beehive instance; change the owner of the command if this is not the case.

Archive and compress the files listed in the file generated by the `beectl clone_preparation` command with your preferred tool for archiving. For example, you can use WinZip on Windows and `tar` and `gzip` on UNIX-based systems. Make sure that the tool you are using preserves the permissions and timestamps of the files.

WARNING: Only archive and compress those files listed in the file generated by the `beectl clone_preparation` command, not the entire Oracle home you want to clone.

The following example shows how to archive and compress the source on Solaris:

```
cd <source Oracle home>
tar -cf - -I <file created by beectl clone_preparation command> | gzip >
clone_image.tar.gz
```

Notes: The `tar` utility may issue warnings if the sticky bit is set on some files. You can safely ignore these warnings.

Do not use the `jar` utility to archive and compress the Oracle home. This avoids warnings or errors from the zip tool about zipping open files (for example, the `<Oracle home>/jdk` files).

Application Tier Cloning

Cloning the application tier consists of the following steps:

- [Step 1: Unzip Compressed Oracle Home](#)
- [Step 2: Set PERL5LIB Environment Variable](#)
- [Step 3: Modify Oracle Home Path](#)
- [Step 4: Execute beectl clone_midtier Command](#)

- [Step 5: Perform Miscellaneous Operations](#)

Step 1: Unzip Compressed Oracle Home

1. Copy the compressed Oracle home from the source machine to the destination machine.
2. Extract the compressed Oracle home into a directory, which will become the new Oracle home at the destination location. Use your preferred tool to extract the compressed files. For example, you can use WinZip on Windows and `tar` and `gunzip` on UNIX-based systems. Make sure that the tool you are using preserves the permissions and timestamps of the files. The following example shows how to extract the files on UNIX-based systems:

```
mkdir -p <destination Oracle home>
cd <destination Oracle home>
gunzip < <directory containing tar file>/clone_image.tar.gz | tar xf -
```

Notes: Make sure that the `tar` and `gzip/gunzip` versions on the source and destination machines are compatible. You may encounter problems unzipping the archive if these versions differ.

The OS (operating system) user doing this **must** have permission to create and update the Oracle inventory. The Oracle inventory is a repository for all Oracle products installed on a host. This is typically a directory named `oraInventory`. The clone operations (described in the sections "[Application Tier Cloning](#)" and "[Site Cloning](#)") will try to create or update the existing Oracle inventory. If the OS user does not have permission, then cloning will fail, and it is not possible to recover from such failure. You will have to retry the cloning procedure.

To determine with OS group has permission to update the Oracle inventory on LinuxSolaris, see the file `/var/opt/oracle/oraInst.loc`. For example:

```
prompt> cat /etc/oraInst.loc/var/opt/oracle/oraInst.loc
inventory_loc=/private/bee hive/oraInventory
inst_group=g900
```

In this example, OS users that belong to group `g900` have permission to update the Oracle inventory, which is located in `/private/bee hive/oraInventory`.

Step 2: Set PERL5LIB Environment Variable

You must have Perl 5.83 or later installed on your system.

Before running the cloning Perl scripts, set the `PERL5LIB` environment variable to the path of the Perl directory in the Oracle home. This path must be the first one listed in the variable definition. For example:

```
setenv PERL5LIB $ORACLE_HOME/perl/lib/5.8.3/sun4-solariosthread-multi:
$ORACLE_HOME/perl/lib/5.8.3:
$ORACLE_HOME/perl/lib/site_perl/5.8.3/sun4-solaris-thread-multi/
```

Step 3: Modify Oracle Home Path

Note: This step applies only to UNIX-based platforms.

The `beectl` command is a Perl script that has the path to Oracle home embedded in it. Modify this path to the new Oracle home. Execute the following `beectl` command to update the embedded Oracle home path. Note that in this case you must add the Perl executable path to the command:

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/beehive/bin/beectl modify_beectl
--new_oracle_home <fully qualified path to new Oracle home>
```

The following table describes the options for the `beectl modify_beectl` command:

Table 24–2 *beectl modify_beectl Options*

Option	Mandatory/ Optional	Description
<code>--new_oracle_home</code>	Mandatory	Path of the new Oracle home. Specify only a fully qualified path without trailing slashes. For example: <code>/app/oracle</code>

Step 4: Execute `beectl clone_midtier` Command

The `clone_midtier` command creates a new application tier and configures it:

```
beectl clone_midtier [options]
```

Note: If cloning fails during this step, you must restart the Oracle Beehive cloning process.

Delete the new Oracle home, and ensure that references to this Oracle home from the Oracle Universal Installer inventory are deleted.

Start the Oracle Beehive cloning process once again, preferably in a different directory Oracle home location.

Note that the `beehive clone_midtier` does not affect the application tier you are trying to clone. Therefore, you do not need to restore this application tier if cloning fails during this step.

The following table describes the options for the `beectl clone_midtier` command:

Table 24–3 *beectl clone_midtier Options*

Option	Mandatory/ Optional	Description
<code>--ias_instance_name</code>	Mandatory	<p>The instance name for the clone.</p> <p>Notes: The instance name should not contain the period (.) character nor the host name itself.</p> <p>The instance name must be different from the source instance and any other instances that use the same Oracle Application Server infrastructure or that are part of the same cluster topology.</p>
<code>--host_name</code>	Mandatory	<p>The hostname of the machine on which the clone is being created. This must be the fully qualified hostname (with the domain appended). For example, <code>hostB.example.com</code>.</p>
<code>--db_schema_password</code>	Mandatory	<p>Database password for the BEE_CODE schema.</p> <p>The password must be the same as the one used during the installation of the source application tier. This is the password of the Oracle Beehive database schema (typically BEE_CODE).</p> <p>Note: If you are not in shell mode, you must obfuscate the database password and add the <code>--obfuscated</code> option to the <code>beectl clone_midtier</code> command.</p> <p>To obfuscate a password, use the <code>beectl obfuscate</code> command:</p> <pre>beectl obfuscate --expiration_time_in_minutes 0 Enter value for password:</pre> <p>Successfully obfuscated the string.</p>
<code>--oui_inv_ptr_loc</code>	Optional	<p>Oracle Universal Installer inventory location.</p> <p>Do not specify if it does not exist on the machine from which you are running this command; in this case, the Oracle inventory will be created in the user's home directory.</p> <p>For more information, refer to "Oracle Inventory Location Option of Clone Commands on UNIX-Based Systems".</p> <p>The Oracle Beehive cloning process internally uses the Oracle Universal Installer to update the Oracle inventory.</p> <p>The value of this option specifies the Oracle Universal Installer inventory location. For example:</p> <pre>--oui_inv_ptr_loc "/var/opt/oracle/oraInst.loc"</pre> <p>Note: This value is platform-dependent. On Solaris it is <code>/var/opt/oracle/oraInst.loc</code>.</p>
<code>--oracle_home_name</code>	Optional	<p>Oracle home name. The default value is the value for the <code>--ias_instance_name</code> option.</p>

Table 24–3 (Cont.) beectl clone_midtier Options

Option	Mandatory/ Optional	Description
--do_not_start_at_end	Optional	<p>If true, Oracle Beehive will not start components after cloning. Permitted value is a boolean value.</p> <p>Setting this option to true will prevent the cloned site from contacting external resources (such as LDAP, virus scanner, voicemail gateway, or Oracle Collaboration Coexistence Gateway (Windows only) of the source site.</p> <p>If you run <code>beectl clone_site</code> for the first time from a source site that has an external source enabled, you will receive a warning message similar to the following:</p> <pre>WARNING : Processing UserDirectoryService : _ UserDirectoryService WARNING : UserDirectoryService is configured with following ENABLED directory profiles WARNING : WARNING : Directory profile id : 880c0691-0d10-4e07-9da0-6d23ab972105 WARNING : LDAP server id : AUTO_DTE_ LDAP_example.com WARNING : LDAP server name : example.com WARNING : LDAP server port : 389 WARNING : LDAP server SSL port : 636</pre> <p>For this example, you would disable your directory profile before continuing.</p>

Step 5: Perform Miscellaneous Operations

Note: This step applies only to UNIX-based platforms.

1. Run the `root.sh` script in the new Oracle home so that the cloned instance works properly. You must log in as the root user to run the script. The script is located in the cloned instance's Oracle home directory, for example: `$ORACLE_HOME/root.sh`.
2. If this is the first Oracle installation on the host, run the `oraInstRoot.sh` script as the root user to register the Oracle inventory directory. The script is located in the `oraInventory` directory.

Site Cloning

Run all the steps described in "[Application Tier Cloning](#)", except call the `beectl clone_site` command (instead of `beectl clone_midtier`).

Step 1: Unzip Compressed Oracle Home

This step is the same as "[Step 1: Unzip Compressed Oracle Home](#)".

Step 2: Set PERL5LIB Environment Variable

This step is the same as ["Step 2: Set PERL5LIB Environment Variable"](#).

Step 3: Modify Oracle Home Path

This step is the same as ["Step 3: Modify Oracle Home Path"](#)

Step 4: Execute beectl clone_site Command

Note: If cloning fails during this step, you must restart the Oracle Beehive cloning process.

Delete the new Oracle home, and ensure that references to this Oracle home from the Oracle Universal Installer inventory are deleted.

Start the Oracle Beehive cloning process once again, preferably in a different directory Oracle home location.

The `beectl clone_site` command creates the first application tier in a site and configures it. This command clears the application tier topology of the old site from the central configuration repository (stored in the database) and creates a new topology for the new site with this as the first and only application tier. It then reconfigures the files on the new Oracle home to work against the new site.

This command is designed and tested so that none of the processes in the new site ever connect to the old site (and vice versa; the old site is ignorant of the new site).

Note: The `beectl clone_site` also performs the following:

- It clears the configuration of Oracle RAC nodes in the central configuration repository. In particular, it clears the properties `XaServiceNames` and `OnsNodeConfiguration` in the database configuration object. Because a new database has been created for the new site, the Oracle RAC configuration for the old database will not be needed.
 - It deletes the configuration of `UnmanagedBeehiveInstance` from the central configuration repository. In particular, the DMZ application tiers and their configurations are deleted.
 - `UnmanagedOc4j` is not deleted.
 - The site name is not changed. You may change the site name with the `beectl modify_property` command.
-
-

The following table describes the options of the `beectl clone_site` command:

Table 24–4 *beectl clone_site Options*

Option	Mandatory/ Optional	Description
--ias_instance_name	Mandatory	The instance name for the clone. Notes: The instance name should not contain the period (.) character nor the host name itself. The instance name must be different from the source instance and any other instances that use the same Oracle Application Server infrastructure or that are part of the same cluster topology.
--host_name	Mandatory	The hostname of the machine on which the clone is being created. This must be the fully qualified hostname (with the domain appended). For example, hostB.example.com.
--db_connect_string	Mandatory	Database connect string for the new site. This would be the connect string for the cloned database.
--db_schema_password	Mandatory	Database password for the schema. Note: If you are not in shell mode, you must obfuscate the database password and add the --obfuscated option to the beectl clone_site command. To obfuscate a password, use the beectl obfuscate command: beectl obfuscate --expiration_time_in_minutes 0 Enter value for password: Successfully obfuscated the string.
--db_schema_name	Optional	New database schema name. Typically, this would be the same schema as the old site, which is usually BEE_CODE.

Table 24–4 (Cont.) *beectl clone_site* Options

Option	Mandatory/ Optional	Description
<code>--db_rac_node_information</code>	Optional	<p>New values for the host:port of Oracle RAC nodes.</p> <p>The host name should not be the VIP hostname. Specify the actual machine name instead.</p> <p>The port should be the ONS remote port, which is also known as the CRS port. This port number is specified in the file <code><Oracle RAC database home>/opmn/conf/ons.config</code>.</p> <p>This option is required to configure ONS properly for Fast Connection Failover, which provides failover for a JDBC connection to a RAC database.</p> <p>This option can be specified more than once and values will form an array in the given order. For example:</p> <pre>--db_rac_node_information "hostnode1.example.com:1521" --db_rac_node_information "hostnode2.example.com:1521" --db_rac_node_information "hostnode3.example.com:1525"</pre>
<code>--db_xa_service_name</code>	Optional	<p>New values for the service names of Oracle RAC nodes. This option can be specified more than once and values will form an array in the given order. For example:</p> <pre>--db_xa_service_name node1_service_name --db_xa_service_name node2_service_name --db_xa_service_name node3_service_name</pre>
<code>--retain_rac_node_information</code>	Optional	<p>Retain existing values for <code>db_xa_service_name</code> and <code>db_rac_node_information</code>. This option cannot be specified with <code>--db_xa_service_name</code> and <code>--db_rac_node_information</code>.</p>
<code>--oracle_home_name</code>	Optional	<p>Oracle home name. The default value is the value provided for the <code>--ias_instance_name</code> option.</p>

Table 24–4 (Cont.) beectl clone_site Options

Option	Mandatory/ Optional	Description
--oui_inv_ptr_loc	Optional. Do not specify if it does not exist on the machine from which you are running this command; in this case, the Oracle inventory will be created in the user's home directory.	Oracle Universal Installer inventory location. For more information, refer to " Oracle Inventory Location Option of Clone Commands on UNIX-Based Systems ". The Oracle Beehive cloning process internally uses the Oracle Universal Installer to update the Oracle inventory. The value of this option specifies the Oracle Universal Installer inventory location. For example: <pre>--oui_inv_ptr_loc "/var/opt/oracle/oraInst.loc"</pre> Note: This value is platform-dependent. On LinuxSolaris it is <code>/etc/oraInst.loc/var/opt/oracle/orInst.loc</code> .
--do_not_start_at_end	Optional	If true, Oracle Beehive will not start components after cloning. Permitted value is a boolean value. Setting this option to true will prevent the cloned site from contacting external resources (such as LDAP, virus scanner, voicemail gateway, or Oracle Collaboration Coexistence Gateway (Windows only) of the source site. If you run <code>beectl clone_site</code> for the first time from a source site that has an external source enabled, you will receive a warning message similar to the following: <pre>WARNING : Processing UserDirectoryService : _ UserDirectoryService WARNING : UserDirectoryService is configured with following ENABLED directory profiles WARNING : WARNING : Directory profile id : 880c0691-0d10-4e07-9da0-6d23ab972105 WARNING : LDAP server id : AUTO_DTE_ LDAP_example.com WARNING : LDAP server name : example.com WARNING : LDAP server port : 389 WARNING : LDAP server SSL port : 636</pre> For this example, you would disable your directory profile before continuing.
--site_name	Optional (Oracle Beehive Release 1 (1.3) and later)	If you specify this option, the site cloning process will clone Oracle Beehive on the target application tier with this new name as the site name of the cloned application tier.

Table 24–4 (Cont.) beectl clone_site Options

Option	Mandatory/ Optional	Description
--ignore_validation_warnings	Optional	If you specify this option, the site cloning process will proceed regardless of warnings about target application tiers referring to external resources. Refer to " Step 5: Prevent Services from Target Application Tiers from Referring to External Resources " for more information about external resources.

Step 5: Prevent Services from Target Application Tiers from Referring to External Resources

In a typical Oracle Beehive installation, Oracle Beehive services may refer to some external resources such as an LDAP server or a virus scan engine. If you perform a site clone of such an installation, the target application tier may also refer to the same external resources.

After running the command `beectl clone_site`, prevent any Oracle Beehive services of the target application tier from referring to the external resources of the source site. Afterwards, you may configure the target application tier to refer to a new set of external resources.

The `beectl clone_site` command will return warning or error messages if the target application tier refers to any external resources. If you do not receive any warning or error messages, proceed to the next step.

If you do receive any warning or error messages from `beectl clone_site` about external resources, stop any Oracle Beehive service from referring to an external resource by following one or more of these steps:

- [Stopping User Directory Service from Referring to LDAP Server](#)
- [Stopping Authentication Service from Referring to LDAP Server](#)
- [Stopping Virus Scanner Process from Referring to External Virus Scan Engine](#)

Notes: Only perform these steps on the target application tier; do not perform these steps on the source site.

These steps are only applicable for cloning a site; do not perform these steps if you are cloning an application tier.

The target application tier may not work as expected during the time between the completion of site cloning process and the modification of Oracle Beehive services to refer to a new set of external resources.

Stopping User Directory Service from Referring to LDAP Server

If you have synchronized User Directory Service (UDS) with an external LDAP server (as described in "[Integrating and Synchronizing LDAP with Oracle Beehive](#)"), then disable all the directory profiles from your target application tier:

1. Retrieve a list of all directory profile objects in your target application tier:

```
beectl list_components --type "UserDirectoryService\${DirectoryProfile}"
```
2. For each directory profile ID, run the following command:

```
beectl modify_property
  --component <directory profile ID>
  --name ProfileState
  --value DISABLE
  --activate_configuration
```

Stopping Authentication Service from Referring to LDAP Server

If you have followed the steps described in "[Configuring Authentication Service to Use LDAP Server](#)", change the authentication mode to use the database instead in your target application tier:

1. Retrieve the Authentication Service ID of your target application tier:

```
beectl list_components --type AuthenticationService
```

2. Change the property AuthStoreType to db:

```
beectl modify_property
  --component <Authentication Service ID>
  --name AuthStoreType
  --value db
  --activate_configuration
```

Stopping Virus Scanner Process from Referring to External Virus Scan Engine

To stop the virus scanner process from referring to an external virus scan engine, remove the reference of VirusScannerCluster from _CURRENT_SITE of your target application tier with the following command:

```
beectl modify_property
  --component _CURRENT_SITE
  --name VirusScanEngineCluster
  --revert_to_default
```

Step 6: Perform Miscellaneous Operations

This step is the same as "[Step 5: Perform Miscellaneous Operations](#)".

OC4J Instance Cloning

OC4J instance cloning allows you to clone Oracle Beehive's managed components, in particular, the OC4J managed components BEEAPP, oc4j_soa, and BEECORE. You may not clone BEEMGMT.

The `beectl clone_oc4j_instance` command clones an Oracle Beehive managed component by creating a new OC4J instance and deploying all the services in the source managed component in the newly cloned OC4J instance. You may only clone Oracle Beehive managed components with this command; you may not clone non-Oracle Beehive OC4J instances.

Note: If you are cloning the BEEAPP managed component, you must backup Oracle Beehive before and after creating the clone.

The BEEAPP clone makes changes to the application tier's configuration files that the `beectl modify_local_configuration_files` command cannot update when restoring Oracle Beehive from a previous backup.

The following table describes the options of the `beectl clone_oc4j_instance` command:

Table 24–5 *beectl clone_oc4j_instance Options*

Option	Mandatory/Optional	Description
<code>--source_oc4j_instance_id</code>	Either this option or <code>--source_oc4j_instance_name</code> is required	ID of the managed component to be cloned, for example, <code>BEEAPP_instance1.example.com</code>
<code>--source_oc4j_instance_name</code>	Either this option or <code>--source_oc4j_instance_id</code> is required	Prefix of the managed component to be cloned, for example <code>BEEAPP</code> , <code>oc4j_soa</code> , or <code>BEECORE</code>
<code>--target_oc4j_instance_name</code>	Mandatory	Name of the new OC4J instance to be created. The application tier instance name and the host name will be appended to this name to create the ID of the new OC4J instance. For example, if you specify <code>BEEAPP_CLONE</code> , its ID will be similar to <code>BEEAPP_CLONE_instance1.example.com</code> .

The following example clones `BEEAPP_instance1.example.com`, which creates a new OC4J instance with the ID `BEEAPP_CLONE_instance1.example.com` and deploys all the services in `BEEAPP` in `BEEAPP_CLONE`:

```
beectl clone_oc4j_instance
  --source_oc4j_instance_name BEEAPP
  --target_oc4j_instance_name BEEAPP_SOURCE
```

Oracle Beehive Integration for Zimbra Cloning

Oracle Beehive Integration for Zimbra is available for Oracle Beehive Release 1 (1.3) and later.

You may clone an instance of Oracle Beehive Integration for Zimbra; follow the steps described in "[Application Tier Cloning](#)" and apply them to the Oracle Beehive Integration for Zimbra home.

Note: Site cloning is only possible for a server application tier, such as Oracle Beehive; it is not possible for a client application tier, such as Oracle Beehive Integration for Zimbra.

Customizing Files or Directories in a Cloned Image

The `beectl clone_preparation` command controls which files or directories are packaged in the clone image, which you may customize.

The `beectl clone_preparation` command refers to the file `<Oracle home>/beehive/conf/scripts/exclude_while_cloning.txt` to obtain the list of files and directories that will be excluded from the clone image.

You may customize this file before executing the `beectl clone_preparation` command.

The comments section in this file describes how to customize it.

Customizing Ports in a Cloned Instance

The `beectl clone_midtier` and `clone_site` commands retain the port values from the source application tier configuration. That is, the cloned application tier will listen on the same ports as the source application tier.

You may customize the ports before executing the `beectl clone_midtier` and `clone_site` commands by updating the file `<Oracle home>/beehive/conf/scripts/clone_ports.ini`. The comments section in this file describes how to override port values.

This file is just an overriding mechanism and does not contain the existing ports, in particular, the ports on which source application tier instance is listening. You may call the `beectl list_ports` command on the source application tier instance to view the existing port assignments, which will also be the port assignments for the cloned application tier instance.

Cloning Application Tiers and Sites with Ports Less Than 1024

If the source application tier or site has been configured with ports less than 1024, and you want to retain the same ports in the cloned application tier or site, follow these steps during the cloning process:

1. Perform "[Step 6: Zip Files to Create Clone Image](#)" as the `root` user. Certain files, such as `<Oracle home>/beehive/bin/hasbind` and `<Oracle home>/Apache/Apache/bin/.apachectl`, require more permissions to be archived properly.
2. Run the `beectl clone_midtier` or `beectl clone_site` commands with the option `--do_not_start_at_end` to ensure that no Oracle Beehive processes are started after a successful cloning. This will prevent some processes from failing because they still have to be configured for privileged ports (less than 1024).
3. Configure the cloned application tier or site by following the steps described in "[Configuring Oracle Beehive to Listen on Ports Less Than 1024](#)".

If you cloned an application tier or a site with ports less than 1024, and you want to reconfigure the ports of the cloned application tier or site to use non-privileged ports (greater than 1024), then follow the steps described in "[Customizing Ports in a Cloned Instance](#)".

Oracle Inventory Location Option of Clone Commands on UNIX-Based Systems

Typically, information about Oracle products on a UNIX-based host are stored in a single location, the Oracle inventory. The location of the Oracle inventory is defined in the Oracle inventory location pointer file. For Solaris, the Oracle inventory location pointer file is `/var/opt/oracle/oraInst.loc`:

```
prompt>cat /var/opt/oracle/oraInst.loc
inventory_loc=/private/beehive/oraInventory
inst_group=g900
```

The `beectl clone_midtier` and `clone_site` commands (with the aid of Oracle Application Server scripts) use the Oracle inventory location pointer from its default location (`/var/opt/oracle/oraInst.loc` on Solaris) to determine the location of

the Oracle inventory. The Oracle inventory is updated with any new Oracle Beehive application tier instance information so that the standard Oracle install and upgrade tools such as Oracle Universal Installer and Opatch will work seamlessly on the cloned application tier instance.

The Oracle inventory location pointer file can be located elsewhere. If this file is not located in the platform default location (`/var/opt/oracle/orainst.loc` on Solaris), then you must specify its location when executing the `beectl clone_midtier` and `clone_site` commands.

Cloned Application Tiers Are Not Automatically SSL or AJPS Enabled

Cloned application tiers are not SSL or AJPS enabled even if the source image is SSL or AJPS enabled.

Enabling SSL and AJPS is a post-install configuration step. Currently, the Oracle Beehive cloning process deliberately does not preserve the SSL and AJPS settings of the source image because the process of enabling them is specific to each application tier and requires administrator input (such as the generation of new certificates).

You must individually enable SSL and AJPS for each of your cloned application tiers. Note that the source application tier is not affected and will remain SSL and AJPS enabled.

For more information about configuring SSL, or more specifically TLS (Transport Layer Security), which is the successor of SSL, and AJPS, refer to the following modules:

- [Configuring TLS with Oracle Wallet](#)
- [Configuring SSL for LDAP Integration](#)
- [Enabling AJPS](#)

Cloned Application Tiers and LDAP Synchronization

If you have cloned an application tier that you have synchronized with an LDAP server, as described in "[Integrating and Synchronizing LDAP with Oracle Beehive](#)", the cloned application tier should still be synchronized with the same LDAP server.

However, if you have configured the Domain Name Service (DNS) on the host of the source instance, as described in "[Active Directory Considerations](#)", you must perform the same configurations on the host of the cloned instance.

If you wish to synchronize your cloned Oracle Beehive instance with another LDAP server, that LDAP server must be a clone of the source LDAP server. It must have matching GUIDs as the source LDAP server, although not necessarily matching hostname, port, or administrator credentials. Refer to the next section, "[Replicating LDAP Server for Cloned Instance](#)" to create a clone of the source LDAP server (these directions are specific to Oracle Internet Directory).

Replicating LDAP Server for Cloned Instance

If you have cloned an application tier that you have synchronized with an LDAP server, the cloned application tier will be synchronized with the same LDAP server. However, if you wish to synchronize the cloned instance with a replicated LDAP server instead, follow the steps in this section.

These steps only apply to a source instance that is synchronized with Oracle Internet Directory.

1. Install a new instance of an LDAP server for the cloned instance.
 - If you are using Oracle Internet Directory, install it in replicated mode. You may choose any type of replication (LDAP replication or Advanced Replication). However, you probably only need one-way LDAP replication; you probably do not want changes in the cloned LDAP server to be propagated to the source LDAP server.

For more information about installing Oracle Internet Directory in replicated mode, refer to Chapter 6, "Installing Oracle Internet Directory in Replicated Mode" in *Oracle Application Server Installation Guide for Solaris Operating System (SPARC 64-Bit)*.
 - If you are using Active Directory, create a new domain controller.
2. Add a replica of the supplier (the source LDAP server) to the LDAP server you just created (which is called the consumer). Replicate the nodes specified in the source instance's LDAP mapping profile (in particular, the DNs specified in <user_search_base> and <groups_search_base>).

- If you are using Oracle Internet Directory, for more information about adding a replica, refer to Chapter 30, "Oracle Internet Directory Replication Installation and Configuration" in *Oracle Internet Directory Administrator's Guide*.
- If you are using Active Directory, create a new replica of the application directory partition and add it to the domain controller you created in the previous step. (In particular, replicate the application directory partitions identified by <user_search_base> and <groups_search_base> in the source instance's LDAP mapping profile, then add those replicas to the domain controller you created in the previous step.)

3. Ensure that the names of the LDAP mapping profiles of the source and cloned instances are the same.
4. If you are using Oracle Internet Database, set the attribute orclDIPRepository to true in your consumer.
5. Retrieve the LDAP mapping profile from the source instance with the following command:

```
beectl list_directory_profiles --file <your home directory>/source_profile.xml
```

The LDAP mapping profile will be saved in the file specified by the `--file` option; in this example, this file is `<your home directory>/source_profile.xml`.

6. Update LDAP mapping profile you just retrieved (`<your home directory>/source_profile.xml`) with values that correspond to the cloned instance and the replicated LDAP server:
 - Set <profile_state> to DISABLE
 - Update the obfuscated <ldap_user_password>. Call the following command on the cloned instance to get a new obfuscated password for the LDAP administrator's password:

```
beectl obfuscate --expiration_time_in_minutes 0
```

- Ensure that the enterprise and organization IDs are correct for the cloned instance.
- Change the SSL and non-SSL port, if required.

7. Delete the existing profile on the cloned instance:

```
beectl delete_directory_profile --profilename "My Profile"
```

Retrieve the name of the existing profile from the <profile_name> element.

8. Add the LDAP mapping profile you modified in step 6 to the cloned instance:

```
beectl add_directory_profile --file ~/source_profile.xml
```

9. Restart the BEECORE and BEEMGMT processes on the cloned instance:

```
beectl status
```

```
-----+-----
Component identifier      | Component type | Status
-----+-----
BTI_instance1.example.com | BTI            | RUNNING
-----+-----
BEEAPP_instance1.example.com | OC4J          | RUNNING
-----+-----
BEEMGMT_instance1.example.com | OC4J          | RUNNING
-----+-----
BEECORE_instance1.example.com | OC4J          | RUNNING
-----+-----
oc4j_soa_instance1.example.com | OC4J          | RUNNING
-----+-----
ohs_instance1.example.com   | HTTP_Server   | RUNNING
-----+-----
```

```
beectl restart --component BEEMGMT_instance1.example.com
beectl restart --component BEECORE_instance1.example.com
```

10. This step is optional. Temporarily disable LDAP authentication with the following beectl commands:

```
beectl modify_property --component _AuthenticationService
  --name AuthStoreType --value db
beectl activate_configuration
beectl modify_local_configuration_files
```

11. Update the BEE_DATA.UDS_SYNC_PROFILE table with the changelog information from the replicated LDAP server:

```
SELECT chg_no FROM ods.ods_chg_log WHERE rownum = 1 ORDER BY chg_no desc;
```

- a. If you are using Oracle Internet Directory, retrieve the change log value from the replicated Oracle Internet Directory by executing the following query on the Oracle Directory Server (ODS) schema:

```
SELECT chg_no FROM ods.ods_chg_log WHERE rownum = 1 ORDER BY chg_no desc;
```

- b. If you are using Active Directory, retrieve the change log value from the domain controller that contains your replicated users and groups with the following command:

```
ldapsearch
  -p <Port of the domain controller>
```

```
-h <Hostname of the domain controller>
-D "<Administrator name of the Active Directory's Windows domain>"
-w "<Administrator Password>"
-b ""
-s base "objectclass=*"
highestCommittedUSN
```

c. Update the BEE_DATA.UDS_SYNC_PROFILE table:

```
UPDATE bee_data.uds_sync_profile
SET changeid = <Value retrieved from previous query>;
```

12. Re-enable LDAP synchronization:

```
beectl modify_property --component <Profile name>
--name ProfileState --value ENABLE
beectl activate_configuration
beectl restart --all
```

13. Re-enable LDAP authentication:

```
beectl modify_property --component _AuthenticationService
--name AuthStoreType --value ldap
beectl activate_configuration
beectl modify_local_configuration_files
```

Testing Replicated LDAP in Cloned Instance

Perform these tasks to ensure that the replicated Oracle Internet Directory server is working in your cloned instance:

1. Create a new user in your source Oracle Internet Directory instance
2. Check your source Oracle Beehive instance; the new user you created should appear in UDS.
3. Check the ODS_CHG_LOG table from the ODS schema from the replicated Oracle Internet Directory instance. You should see your changes; expect a delay of one hour.
4. Between fifteen to thirty minutes later, you should see a change in the cloned Oracle Beehive instance's UDS.

Troubleshooting Replicated LDAP

- Check the log files from the BEEMGMT and BEECORE processes.
- For more troubleshooting tips, refer to the section "[Troubleshooting Synchronization between Oracle Beehive and Oracle Internet Directory](#)" in "[Integrating and Synchronizing LDAP with Oracle Beehive](#)".

Site Cloning and Multiple Instances

Oracle Beehive site cloning will result in a single application tier in the cloned site irrespective of the number of application tiers in the source site. To create more application tiers in the cloned site, follow the procedures described in "[Application Tier Cloning](#)" in the cloned site. Note that a cloned application tier will not be SSL enabled even if the source image is SSL enabled. Refer to "[Cloned Application Tiers Are Not Automatically SSL or AJP/SSL Enabled](#)" for more information.

References to Oracle Application Server Cloning Documentation

Oracle Beehive cloning scripts internally use Oracle Application Server cloning scripts to clone Oracle Application Server components such as OC4J on which Oracle Beehive is based. Refer to the following sections in Chapter 9, "Cloning Application Server Middle-Tier Instances" in *Oracle Application Server Administrator's Guide*:

- Section 9.4.4, Locating and Viewing Log Files
- Section 9.5, Considerations and Limitations for Cloning

Oracle Beehive Administration Console Post-Installation Procedures

Oracle Beehive Administration Console is available for Oracle Beehive Release 1 (1.3) and later.

Depending on your security requirements or any other issue particular to your deployment, perform one or more of the following procedures after installing Oracle Beehive Administration Console:

- [Configuring Oracle Beehive Administration Console for SSL Access](#)
- [Changing Oracle Beehive Administration Console Port](#)
- [Extending Java Single Sign-On Session Timeout](#)
- [Configuring Oracle Beehive Administration Console for LDAP-Based Authentication](#)
- [Configuring Virtual Host](#)
- [Restarting OC4J](#)

Configuring Oracle Beehive Administration Console for SSL Access

1. Create a keystore with the following command. The keytool command is located in *<Oracle Beehive Administration Console home>/jdk/bin*. The following example will create a keystore named *server.keystore* in the directory *<Oracle Beehive Administration Console home>* with the password *welcome*:

```
keytool -genkey -keyalg "RSA"  
-keystore <Oracle Beehive Administration Console home>/server.keystore  
-storepass welcome -validity 90
```

Note: For more information about the keytool utility, refer to <http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html>.

For additional information, refer to "Using Keys and Certificates with OC4J and Oracle HTTP Server" and "Using SSL with Standalone OC4J" in Chapter 15, "SSL Communication with OC4J" in *Oracle Containers for J2EE Security Guide*.

2. Make a copy of *<Oracle Beehive Administration Console home>/j2ee/home/config/default-web-site.xml* and name it *<Oracle*

Oracle Beehive Administration Console
home>/j2ee/home/config/secure-web-site.xml.

3. In `secure-web-site.xml`, edit the `<web-site>` element. Set the `port` attribute to any available port number, `secure` to `true`, and `protocol` to `https`:

```
<web-site
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "http://xmlns.oracle.com/oracleas/schema/11/web-site-11_1.xsd"
  port="4443"
  secure="true"
  protocol="https"
  display-name="Default Web Site"
  schema-major-version="11"
  schema-minor-version="1">
```

Note: You may also add the attribute `virtual-hosts` to the `<web-site>` to use both SSL and virtual hosts. Refer to ["Configuring Virtual Host"](#) for more information.

4. Add the following element inside the `<web-site>` element:

```
<ssl-config
  keystore="<Oracle Beehive Administration Console home>/server.keystore"
  keystore-password="welcome"/>
```

Set the attribute `keystore` to the full path name of the keystore you created previously. Set the attribute `keystore-password` to the password of the keystore.

5. In `<Oracle Beehive Administration Console home>/j2ee/home/config/server.xml`, edit the `<web-site>` element and add the attribute `path=" ./secure-web-site.xml "`:

```
<web-site path=" ./secure-web-site.xml "/>
```

6. Restart OC4J as described in ["Restarting OC4J"](#).

Changing Oracle Beehive Administration Console Port

1. Edit the file `<Oracle Beehive Administration Console home>/j2ee/home/config/default-web-site.xml` and modify the `port` attribute in the `<web-site>` element to any available port number:

```
<web-site ... port="7779" ... >
```

2. Restart OC4J as described in ["Restarting OC4J"](#).

Extending Java Single Sign-On Session Timeout

To extend the Java Single Sign-On session timeout for Oracle Beehive Administration Console, add the highlighted text to the file `<Oracle Beehive Administration Console home>/j2ee/home/application-deployments/javasso/jps-config.xml`:

```
<serviceInstance name="idm" provider="idm.provider">
  <description>JSSO Authentication Configuration</description>
```

```

<property name="idm.authentication.name" value="JavaSSO"/>
<property name="idm.token.assertter.class"
  value="oracle.security.jps.internal.jsso.SSOCookieTokenAssertter"/>
<property name="idm.token.collector.class"
  value="oracle.security.jps.internal.jsso.SSOCookieTokenCollector"/>
<property name="idm.token.type" value="COOKIE_TOKEN"/>
<property name="idm.token.collector.cookie.1" value="ORA_OC4J_SSO"/>
<property name="custom.sso.url.login" value="/jsso/SSOLogin"/>
<property name="custom.sso.url.logout" value="/jsso/SSOLogout"/>
<property name="custom.sso.cred.key" value="JSSO_KEY"/>
<property name="custom.sso.cred.alias" value="JSSO_ALIAS"/>
<property name="custom.sso.session.timeout" value="3600"/>
</serviceInstance>

```

After making changes to the `jps-config.xml` file, restart OC4J as described in ["Restarting OC4J"](#).

For more information about Java Single Sign-On, refer to Chapter 14, "OC4J Java Single Sign-On" in *Oracle Containers for J2EE Security Guide*.

Configuring Oracle Beehive Administration Console for LDAP-Based Authentication

You may configure Oracle Beehive Administration Console so that it authenticates its users with credentials stored in your LDAP directory. These steps apply only if you have integrated and synchronized your Oracle Beehive instance with Oracle Internet Directory. Refer to ["Integrating and Synchronizing LDAP with Oracle Beehive"](#) for more information.

1. Edit the file `<Oracle Beehive Administration Console home>/j2ee/home/application-deployments/javasso/jps-config.xml` and update the values that are highlighted in the following excerpt with configuration information that corresponds to your LDAP directory:

```

<serviceInstance name="beehive.ldap.loginmodule"
  provider="jaas.login.provider">
  <description>Beehive LDAP Login Module</description>
  <property name="loginModuleClassName"
    value="oracle.ocs.csi.authentication.login.modules.OcsLdapLoginModule"/>
  <property name="jaas.login.controlFlag" value="REQUIRED"/>
  <property name="debug" value="true"/>
  <property name="addAllRoles" value="true"/>
  <property name="oracle.security.jaas.ldap.user.object.class"
    value="inetOrgPerson"/>
  <property name="oracle.security.jaas.ldap.provider.connect.pool"
    value="true"/>
  <property name="oracle.security.jaas.ldap.provider.credential"
    value="!welcome1"/>
  <property name="oracle.security.jaas.ldap.provider.type" value="other"/>
  <property name="oracle.security.jaas.ldap.provider.url"
    value="ldap://ldapserver:389"/>
  <property name="oracle.security.jaas.ldap.role.searchscope" value="subtree"/>
  <property name="oracle.security.jaas.ldap.user.searchscope" value="subtree"/>
  <property name="oracle.security.jaas.ldap.user.searchbase"
    value="cn=Users,dc=us,dc=oracle,dc=com"/>
  <property name="oracle.security.jaas.ldap.role.searchbase"
    value="cn=Users,dc=us,dc=oracle,dc=com"/>
  <property name="oracle.security.jaas.ldap.role.object.class"
    value="groupOfUniqueNames"/>

```

```

<property name="oracle.security.jaas.ldap.role.name.attribute" value="cn"/>
<property name="oracle.security.jaas.ldap.provider.user"
  value="cn=orcladmin"/>
<property name="oracle.security.jaas.ldap.user.name.attribute" value="uid"/>
<property name="oracle.security.jaas.ldap.membership.searchscope"
  value="direct"/>
<property name="oracle.security.jaas.ldap.member.attribute"
  value="uniqueMember"/>
<property name="oracle.security.jaas.ldap.lm.cache_enabled" value="true"/>
<property name="authenticated.role.name" value="authenticated-role"/>
</serviceInstance>

```

Tips: Obtain values for `jps-config.xml` from the file `<Oracle Beehive home>/j2ee/home/config/system-jazn-data.xml`. Look for the `<application>` element that contains the element `<name>beehive-auth-framework-password</name>`.

Prepend the clear-text password (the value of `oracle.security.jaas.ldap.provider.credential` of the administrator's account (the value of `oracle.security.jaas.ldap.provider.user`) with an exclamation point (!). The exclamation point will obfuscate the password the next time OC4J is restarted.

2. In the same file (`jps.config.xml`), update the `<jpsContexts>` element with the highlighted value:

```
<jpsContexts default="jss0-ldap">
```

3. Edit the file `<Oracle Beehive Administration Console home>/j2ee/home/application-deployments/myapp/jps-config.xml`, make the same changes as in steps 1 and 2.
4. Restart OC4J as described in ["Restarting OC4J"](#).

Configuring Virtual Host

1. Add `virtual-hosts="beehive_admin.us.oracle.com"` to the `<web-site>` element in `default-web-site.xml`. (Replace `beehive_admin.us.oracle.com` with the name of your virtual host.)

```

<web-site
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "http://xmlns.oracle.com/oracleas/schema/11/web-site-11_1.xsd"
  port="7778"
  secure="false"
  protocol="http"
  display-name="Default Web Site"
  schema-major-version="11"
  schema-minor-version="1"
  virtual-hosts="beehive_admin.us.oracle.com">

```

2. Restart OC4J as described in ["Restarting OC4J"](#).

Restarting OC4J

For information about starting and stopping OC4J, refer to Chapter 5, "Starting and Stopping OC4J" in *Oracle Containers for J2EE Configuration and Administrator's Guide*.

Part III

Oracle Beehive High Availability Configuration

This part describes how to install and configure Oracle Beehive in a high availability environment. It contains the following modules:

- [Configuring and Installing Oracle Beehive for Oracle RAC](#)
- [Installing Oracle Beehive in High Availability Environment](#)

Configuring and Installing Oracle Beehive for Oracle RAC

This module describes how to install and configure Oracle Beehive to support deployment on multiple Oracle Real Application Clusters (RAC) nodes. It covers the following topics:

- [Naming Conventions](#)
- [Pre-Install Steps](#)
- [Installation](#)
- [Post-Install Steps](#)
- [Adding Oracle RAC Nodes to an Existing Oracle Beehive Instance](#)
- [Installing Multiple Oracle Beehive Instances with Oracle RAC](#)

Note: You may use Database Configuration Assistant (DBCA) to create Oracle RAC nodes. However, refer to **OracleMetaLink** Note 342419.1 to properly configure the LOCAL_LISTENER initialization parameter.

Naming Conventions

This module uses the following naming conventions:

- Global service name of the Oracle RAC database to be used: MYDB
- Number of Oracle RAC nodes: R
- SID of first Oracle RAC node: MyDB1
- SID of second Oracle RAC node: MyDB2
- SID of Rth Oracle RAC node: MyDBR
- Virtual Internet Protocol (VIP) host names and listener port numbers for each Oracle RAC node: HOST-VIP1 : PORT1 to HOST-VIPR : PORTR
- Oracle Notification Services (ONS) host names and remote port numbers for each Oracle RAC node: ONS_HOST1 : ONS_PORT1 to ONS_HOSTR : ONS_PORTR

Note: The location of the archive redo logs for each Oracle RAC node must be accessible by all nodes in your Oracle RAC database.

Refer to "Managing Archived Redo Logs Using RMAN in Oracle Real Application Clusters" in Chapter 5, "Configuring Recovery Manager And Archiving" in *Oracle Real Application Clusters Administration and Deployment Guide* for more information.

Pre-Install Steps

These steps consist of the following:

1. [Create Multiple Database Services](#)
2. [Start All Database Services](#)
3. [Enable DTP for Database Services](#)
4. [Enable Secure ONS Notification](#)
5. [Manually Create ORABPEL Tablespace](#)

Step 1 Create Multiple Database Services

At any given time, the database service connects to only one instance of Oracle RAC. For proper load balancing, create multiple logical database services.

The number of database services to create should be between three times the number of Oracle RAC nodes to four times the number of Oracle RAC nodes.

In the following example, N is the number of database services to create. The names of these database services are `Service1`, `Service2`, to `ServiceN`.

The following command creates a database service for an Oracle RAC database with three nodes:

```
srvctl add service -d MYDB -s Service1 -r MyDB1 -a MyDB2, MyDB3
```

`Service1` runs on `MyDB1`. If `MyDB1` fails, the service will be failover to `MyDB2`. Do not use the same Oracle RAC node as the primary node for all the services. Use different RAC nodes as primary nodes and secondary nodes for proper load balancing in failover. The following examples demonstrate this:

```
srvctl add service -d MYDB -s Service2 -r MyDB2 -a MyDB3, MyDB1
srvctl add service -d MYDB -s Service3 -r MyDB3 -a MyDB1, MyDB2
srvctl add service -d MYDB -s Service4 -r MyDB1 -a MyDB3, MyDB2
```

Step 2 Start All Database Services

Start all database services that you created. The following command starts `Service1`:

```
srvctl start service -d MYDB -s Service1
```

Step 3 Enable DTP for Database Services

Enable X/Open Distributed Transaction Processing (DTP) for all database services. The following command enables DTP for `Service1`. Run this command in SQL*Plus with system privileges:

```
execute dbms_service.modify_service (service_name => 'Service1', dtp => true);
```

Step 4 Enable Secure ONS Notification

You may ignore this step if your deployment does not require SSL notification.

Note: You must perform this step if you wish to install Oracle Beehive with Oracle Beehive Provisioning Application. Refer to ["Installing Oracle Beehive with Provisioning Application"](#) for more information.

This step involves specifying the Oracle Wallet directory `ons.config`. As a result Oracle Cluster Ready Services (CRS) will use SSL when communicating with other nodes (other Oracle RAC nodes and Oracle Beehive instances) and require SSL certificate authentication from all Oracle Notification Services (ONS) instances that try to connect to it.

To enable SSL ONS notification, add the following line to all `<Oracle Cluster Ready Services home>/opmn/conf/ons.config` files of each Oracle RAC node:

```
walletfile=MY_SECURE_WALLER_DIR
```

`MY_SECURE_WALLER_DIR` is your Oracle Wallet directory. Refer to ["Configuring TLS with Oracle Wallet"](#) for information about configuring Oracle Wallet.

Step 5 Manually Create ORAPEL Tablespace

If your Oracle RAC database uses raw storage, you must manually create the tablespace ORAPEL before running the Oracle Beehive Install Wizard. To do this, run the following command:

```
CREATE TABLESPACE ORAPEL DATAFILE '/dev/raw/raw1' SIZE 100M
  AUTOEXTEND ON NEXT 30M MAXSIZE UNLIMITED;
```

`/dev/raw/raw1` is the name of the data file to create for the ORAPEL tablespace.

Installation

Install Oracle Beehive with the Oracle Beehive Install Wizard. Enter the following database information:

- **Host and Port:** `HOST-VIP1:PORT1^HOST-VIP2:PORT2^HOST-VIP3:PORT3`
- **Service Name:** `Service1`. Use the database service name you created in the step [Step 1, "Create Multiple Database Services"](#). (Do not use the actual service name. For example, do not use `MYDB`, use `Service1`.)

Note: During the installation of Oracle Beehive, ensure the first Oracle RAC node (`MyDB1`) remains up. Otherwise, the install will fail due to bug 5693566 (Web Services Manager Configuration Assistant will fail).

Post-Install Steps

These steps update Oracle Beehive with the Oracle RAC global service name and the multiple database service names:

1. [Backup Bootstrap File](#)
2. [Retrieve Site and Database Object System Identifiers](#)
3. [Modify Connect String](#)
4. [Update Bootstrap File with Connect String](#)

5. [Update Database System Object](#)
6. [Commit Configuration Changes](#)
7. [Restart Oracle Beehive](#)
8. [Register for ONS Notification](#)

Step 1 Backup Bootstrap File

The file `<Oracle_home>/beehive/conf/beehiveconfig.xml` contains the bootstrap database connect information. Make a backup of this file before applying any post-install steps.

Step 2 Retrieve Site and Database Object System Identifiers

Retrieve the following system identifiers. You will need them for the following steps:

- **Site System Identifier:** Use the following `beectl` command to retrieve the Oracle Beehive site system identifier. In the following steps, `<site_id>` represents this value:

```
beectl list_components --type Site
```

- **Database Object System Identifier:** Use the following `beectl` command to retrieve the default database system identifier. In the following steps, `<database_system_object_id>` represents this value:

```
beectl list_properties --component <site_id>
```

This command lists all the properties of `<site_id>`. The value of `<database_system_object_id>` is the value of the property Database.

Step 3 Modify Connect String

1. List the properties of `<database_system_object_id>` with the following command:

```
beectl list_properties --component <database_system_object_id>
```

2. Change the value of the ConnectDescriptor property. Substitute the database service name (in this example, `Service1`) with the global service name (`MYDB`). For example, suppose the following is the value of the ConnectDescriptor property (line breaks and extra spaces added for clarity):

```
(DESCRIPTION=
  (ADDRESS_LIST=
    (LOAD_BALANCE=yes) (FAILOVER=on)
    (ADDRESS= (PROTOCOL=TCP) (HOST=HOST-VIP1) (PORT=PORT1))
    (ADDRESS= (PROTOCOL=TCP) (HOST=HOST-VIP2) (PORT=PORT2))
  )
  (CONNECT_DATA= (SERVER=DEDICATED) (SERVICE_NAME=Service1))
)
```

You would change it to the following:

```
(DESCRIPTION=
  (ADDRESS_LIST=
    (LOAD_BALANCE=yes) (FAILOVER=on)
    (ADDRESS= (PROTOCOL=TCP) (HOST=HOST-VIP1) (PORT=PORT1))
    (ADDRESS= (PROTOCOL=TCP) (HOST=HOST-VIP2) (PORT=PORT2))
  )
  (CONNECT_DATA= (SERVER=DEDICATED) (SERVICE_NAME=MYDB))
)
```

In the following steps `<modified_connect_string>` will represent this value.

Step 4 Update Bootstrap File with Connect String

Use the following command to update the database system object and the local bootstrap file with the connect string you modified in the previous step:

```
beectl modify_bootstrap_configuration --connect_string "<modified_connect_string>"
```

Note: You must enclose the connect string with quotation marks.

Step 5 Update Database System Object

Modify the database system object's configuration to specify the database services you created and the ONS remote ports that are listening on your Oracle RAC nodes. The following example specifies three database services with the `xa_service_names` option, and two ONS remote ports with the `ons_entry` option:

```
beectl modify_database
  --database <database_system_object_id>
  --xa_service_names Service1
  --xa_service_names Service2
  --xa_service_names Service3
  --ons_entry ONS_HOST1:ONS_PORT1
  --ons_entry ONS_HOST2:ONS_PORT2
```

Step 6 Commit Configuration Changes

Call `beectl activate_configuration` on your Oracle Beehive instance. Afterwards, call `beectl modify_local_configuration_files` on the same Oracle Beehive instance.

Note: The `beectl modify_local_configuration_files` command will ask you to run this command on all your other Oracle Beehive instances. **Do not run this command on all your other instances at this time.**

If you have, or plan to have, more than Oracle Beehive instance, refer to "[Installing Multiple Oracle Beehive Instances with Oracle RAC](#)".

Step 7 Restart Oracle Beehive

Restart your Oracle Beehive instance. You may use the `beectl restart --all` command.

Note: You must restart Oracle Beehive (after calling `beectl modify_local_configuration`) if you modify the ONS details in the database system object's configuration (as you have done previously).

Step 8 Register for ONS Notification

By default, Oracle Beehive is configured to receive secure ONS notifications, which is recommended. If your Oracle RAC nodes are configured for SSL ONS notification, you may ignore this step.

If not, update the file `<Oracle home>/opmn/conf/opmn.xml` as follows in all Oracle Beehive instances to disable SSL ONS notifications:

In the file `opmn.xml`, in the `ssl` element, set the `enabled` attribute to `false`:

```
<ssl enabled="false" wallet-file="$ORACLE_HOME/opmn/conf/ssl.wlt/default"/>
```

After changing the `opmn.xml` file, run the following `beectl` commands:

```
beectl list_components --type OpmnCluster
```

```
-----+-----
Component type | Component identifier
-----+-----
OpmnCluster    | 5987aa04-2984-43ab-8752-8c27e0c45130
-----+-----
```

1 Record(s) displayed.

```
beectl list_properties --component 5987aa04-2984-43ab-8752-8c27e0c45130
```

```
-----+-----
Property name  | Property value
-----+-----
Alias          |
-----+-----
NotificationServerSslEnabled | true
-----+-----
Opmnns        | opmn_Beehive_R1.example.com
-----+-----
Site          | _CURRENT_SITE
-----+-----
```

4 Record(s) displayed.

```
beectl modify_property
```

```
--component 5987aa04-2984-43ab-8752-8c27e0c45130  
--name NotificationServerSslEnabled --value false
```

Changes to configuration repository are not activated.
Successfully stored the property for component id
5987aa04-2984-43ab-8752-8c27e0c45130.

```
beectl activate_configuration
```

Local configuration files are not in sync with system model. Please run
"modify_local_configuration_files" manually.
Proposed configuration is saved successfully and activated now.

```
beectl modify_local_configuration_files
```

Adding Oracle RAC Nodes to an Existing Oracle Beehive Instance

The following example adds an Oracle RAC node named `MyDB3` to an Oracle Beehive instance with the following characteristics:

- Contains two Oracle RAC nodes named `MyDB1` and `MyDB2`
- Contains six database services named `Service1` through `Service6`

- For *Service1*, *Service3*, and *Service5*, the preferred database instance is *MyDB1* and the available database instance is *MyDB2*
 - For *Service2*, *Service4*, and *Service6*, the preferred database instance is *MyDB2* and the available database instance is *MyDB1*
1. Shutdown *MyDB2* and CRS.
 2. Start CRS. Start ASM instances, the node applications, and the instance of *MyDB3*.
 3. For *Service1*, *Service3*, and *Service5*, change the available database instance from *MyDB2* to *MyDB3*:


```

srvctl modify service -d MYDB -s Service1 -n -i MyDB1 -a MyDB3
srvctl modify service -d MYDB -s Service3 -n -i MyDB1 -a MyDB3
srvctl modify service -d MYDB -s Service5 -n -i MyDB1 -a MyDB3

```
 4. For *Service2*, *Service4*, and *Service6*, change the available database instance from *MyDB2* to *MyDB3*:


```

srvctl modify service -d MYDB -s Service2 -n -i MyDB3 -a MyDB1
srvctl modify service -d MYDB -s Service4 -n -i MyDB3 -a MyDB1
srvctl modify service -d MYDB -s Service6 -n -i MyDB3 -a MyDB1

```
 5. Start the databases that have stopped (*Service1*, *Service3*, and *Service5*).

Installing Multiple Oracle Beehive Instances with Oracle RAC

Oracle Beehive supports the following ways to install multiple instances Oracle Beehive with Oracle RAC:

- [Install All Oracle Beehive Instances, then Apply Post-Install Steps](#)
- [Install One Oracle Beehive Instance, Apply Post-Install Steps, then Add Additional Instances](#)

Install All Oracle Beehive Instances, then Apply Post-Install Steps

You can install multiple Oracle Beehive instances, then run the post-install steps afterwards on all those instances as described in this module:

1. Install all your Oracle Beehive instances. When entering the database information for each instance, use the same service name.
2. For one instance, apply all the post-installation steps.
3. For all the other instances, run the following post-installation steps:
 - [Step 4, "Update Bootstrap File with Connect String"](#)
 - [Step 5, "Update Database System Object"](#)
 - [Step 6, "Commit Configuration Changes"](#)
 - [Step 8, "Register for ONS Notification"](#)

Install One Oracle Beehive Instance, Apply Post-Install Steps, then Add Additional Instances

Alternatively, instead of installing multiple Oracle Beehive instances then running the post-install steps on each one, you may install one Oracle Beehive instance, apply the post-install steps on that instance, then add additional Oracle Beehive instances.

After installing one Oracle Beehive instance and applying post-install steps, follow these steps:

1. Install other Oracle Beehive instances. When entering the database information, use the global service name as the server name.
2. Apply [Step 8, "Register for ONS Notification"](#) to all new instances.

Installing Oracle Beehive in High Availability Environment

Installing Oracle Beehive in a high availability environment involves installing a third-party load balancer or an Oracle Beehive DMZ instance (or both) and configuring the virtual server of each Oracle Beehive (non-DMZ) instance. Configuring the virtual server ensures that there is a single point of access to your Oracle Beehive instances.

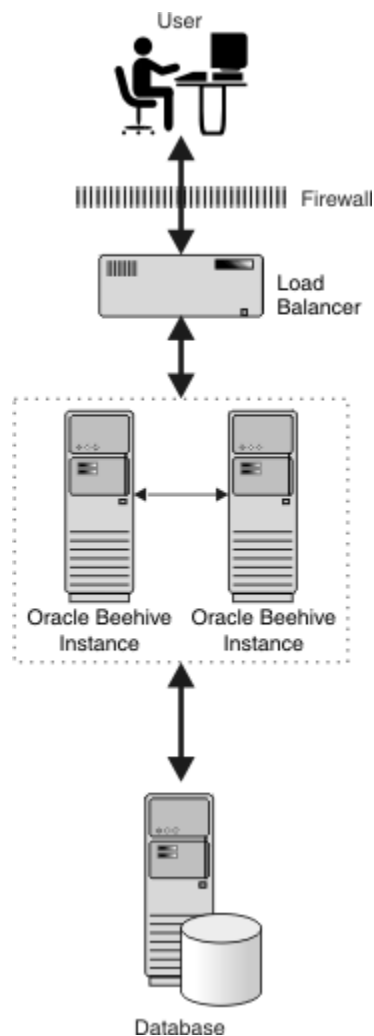
This module covers the following topics:

- [Configuring High Availability Environment with Load Balancer](#)
- [Configuring High Availability Environment with a DMZ Instance](#)
- [Configuring High Availability Environment with DMZ Instances and Load Balancer](#)

Configuring High Availability Environment with Load Balancer

The following diagram illustrates multiple Oracle Beehive instances and a load balancer in a high availability environment:

Figure 27-1 Multiple Instances in High Availability Environment



Follow these steps to install and configure more than one Oracle Beehive instance with a load balancer:

1. Install an Oracle Beehive instance.
2. Change the `ServerName` property of the Oracle Beehive virtual server to the load balancer server name. In the following example, `load-balancer.example.com` is the server name of the load balancer:

```
beectl modify_property --component _VIRTUAL_SERVER
--name ServerName
--value load-balancer.example.com
```

3. If the `HttpPort` property value of the Oracle Beehive virtual server and the HTTP port of the load balancer are different, then change the `HttpPort` property value of the virtual server to the HTTP port of the load balancer. In the following example, the HTTP port of the load balancer is 80:

```
beectl list_properties --component _VIRTUAL_SERVER
-----
| Property Name          | Property Value      |
-----
```

```

| ServerName          | example.com          |
| ImapPort            | 143                  |
| Smtpport            | 25                   |
| XmppPort            | 5222                 |
| IPAddress           |                      |
| HttpPort           | 7777               |
| HttpSslPort         | 4443                 |
| HttpSslEnabled      | true                 |
| FtpPort             | 2121                 |
| BtiClientPort       | 21401                |
| BtiSecureClientPort | 21451                |
| Alias               | _VIRTUAL_SERVER     |
-----

```

```

beectl modify_property --component _VIRTUAL_SERVER
  --name HttpPort
  --value 80

```

4. Commit configuration changes:

```

beectl activate_configuration
beectl modify_local_configuration_files

```

Note: The `beectl modify_local_configuration_files` command will ask you to run this command on all your other Oracle Beehive instances. **Do not run this command on all your other instances at this time.**

5. Ensure that the changes you made in steps 2 and 3 appear in the file `<Oracle home>/Apache/Apache/conf/httpd.conf`.
6. Install another Oracle Beehive instance.

Oracle Beehive will apply these changes to any subsequent Oracle Beehive installation, so you do not have to run `beectl modify_local_configuration_files` on these new instances.

Note: If you are making these changes after installing two or more Oracle Beehive instances, then follow steps 1-5 on the first instance. Afterwards, only run `beectl modify_local_configuration_files` on the other (previously installed) instances.

Configuring High Availability Environment with a DMZ Instance

If you do not have a load balancer, you may use an Oracle Beehive DMZ instance as a load balancer and the single point of access to your Oracle Beehive (non-DMZ) instances. The configuration of this high availability environment is similar to configuring one for a load balancer:

1. Configure one Oracle Beehive DMZ instance as described in "[Configuring Oracle Beehive Demilitarized Zone Instances](#)".
2. Perform the steps described in "[Configuring High Availability Environment with Load Balancer](#)" for your Oracle Beehive (non-DMZ) instances. This involves configuring the virtual server with the load balancer for each Oracle Beehive instance. However, set the `ServerName` property to the server name of your Oracle

Beehive DMZ instance and the `HttpPort` property to the HTTP port of your Oracle Beehive DMZ instance.

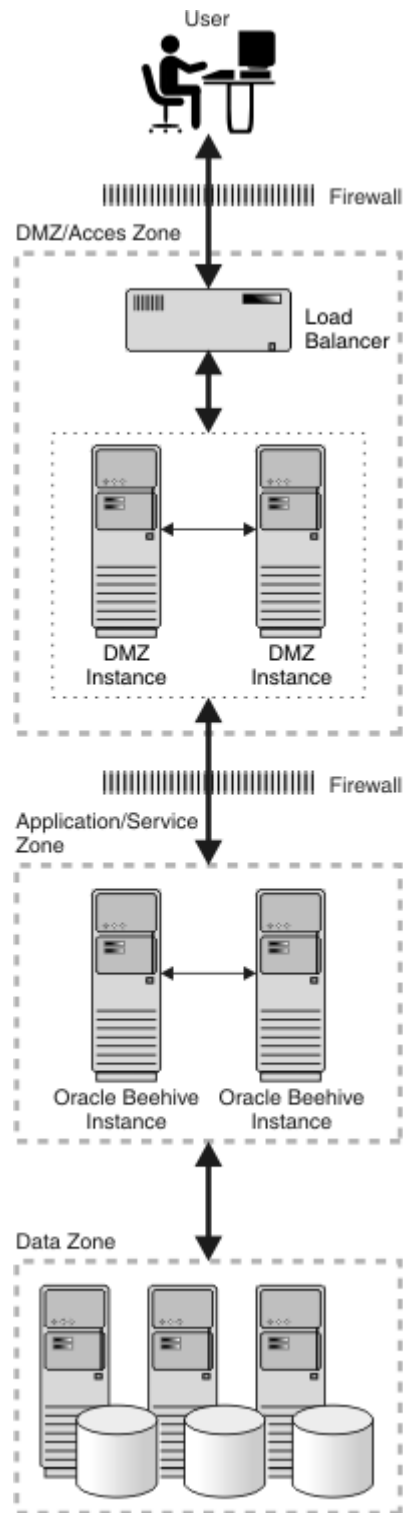
Note that if you are performing these steps after you have installed all your Oracle Beehive instances, run `beectl modify_local_configuration_files` only after configuring the virtual server for all your Oracle Beehive instances.

Configuring High Availability Environment with DMZ Instances and Load Balancer

You may have a high availability environment with both a load balancer and DMZ instances. In this case, your Oracle Beehive instances will be load balanced on multiple levels.

The following diagram illustrates multiple DMZ instances and a load balancer in a high availability environment:

Figure 27-2 Multiple DMZ Instances in High Availability Environment



Follow these steps to configure a high availability environment with a load balancer and multiple Oracle Beehive DMZ instances:

1. Perform the steps described in "[Configuring High Availability Environment with Load Balancer](#)" for your Oracle Beehive (non-DMZ) instances. This involves

configuring the virtual server with the load balancer for each Oracle Beehive instance.

Note that if you are performing these steps after you have installed all your Oracle Beehive instances, run `beectl modify_local_configuration_files` only after configuring the virtual server for all your Oracle Beehive instances.

2. Edit the following directives in the file `<Oracle Beehive DMZ home>/Apache/Apache/conf/httpd.conf`:

```
ServerName <Domain name server (DNS) of the load balancer>  
Port <HTTP port of the load balancer>
```

For example, if `load-balancer.example.com` is the load balancer's DNS, and 80 is its HTTP port number, then edit the `httpd.conf` file as follows:

```
ServerName load-balancer.example.com  
Port 80
```

3. Perform this step only if you need to configure your DMZ instance with SSL.

Retrieve the location of the file `ssl.conf` from the file `<Oracle home>/Apache/Apache/conf/httpd.conf`. Edit the following directives in the `ssl.conf` file:

```
ServerName <Domain name server (DNS) of the load balancer>  
Port <HTTPS port of the load balancer>
```

For example, if `load-balancer.example.com` is the load balancer's DNS, and 4443 is its HTTPS port number, then edit the `ssl.conf` file as follows:

```
ServerName load-balancer.example.com  
Port 4443
```

4. Restart the Oracle Beehive DMZ instance.
5. Repeat steps 1-3 for each of your Oracle Beehive DMZ instances.

Troubleshooting Oracle Beehive Installation

Unable to Perform FTP Operations

Check the log files of BEEAPP, BEECORE, and BEEMGMT. If you see an exception thrown by `oracle.ocs.omb.transport.exception.TimedOutException`, ensure the port defined by `ServerPort` in the BTI component is accessible.

This may occur if you have two Oracle Beehive instances, and you block this port (through your firewall) between these instances.

ORA-12850: Could not allocate slaves on all specified instances, ORA-00018: Maximum number of sessions exceeded

If you receive the error "Failed to access configuration repository (database). Internal error message: `java.sql.SQLException: Connection Unavailable`" when running a `beectl` command and in your database alert log see the errors ORA-12850: Could not allocate slaves on all specified instances and ORA-00018: Maximum number of sessions exceeded, increase the processes initialization parameter in your database. Refer to "[Initialization Parameters](#)" in "[Oracle Beehive Database Requirements](#)" for information on changing initialization parameters in your database.

Oracle Beehive Install Wizard Fails Because Apache HTTP Server Unable to Start

Oracle Beehive Install Wizard may fail because of an error similar to the following:

```
<ORACLE_HOME>/Apache/Apache/bin/apachectl startssl: execing httpd
Syntax error on line 233 of <ORACLE_HOME>/Apache/Apache/conf/httpd.conf:
Cannot load <ORACLE_HOME>/Apache/Apache/libexec/mod_auth_dbm.so into server:
<ORACLE_HOME>/Apache/Apache/libexec/mod_auth_dbm.so: undefined symbol: dbm_fetch
```

The undefined symbol error may result from the symbolic link `/usr/lib/libdb.so.2` pointing to a different file or an incorrect version of the shared library.

Before starting the Install Wizard, create (or recreate) the following symbolic link as the root user:

```
ln -s /usr/lib/libgdbm.so.2.0.0 /usr/lib/libdb.so.2
```



Index

A

Active Directory
 considerations, 14-20
adding to existing instances, 26-6
AJPS
 See Apache JServ Protocol
Apache JServ Protocol, 18-1

B

Bash shell
 setting shell limits, 1-3
beadmin, 4-10
beectl, 12-1
Bourne shell
 setting shell limits, 1-3

C

C shell
 setting shell limits, 1-3
certificate authority, 17-1
cloning, 1-10, 24-1
 application tier, 24-6
 customizing files or directories, 24-17
 customizing ports, 24-18
 LDAP synchronization, 24-19
 multiple instances, 24-22
 OC4J, 24-16
 other documentation, 24-23
 preparing source application tier, 24-4
 replicating LDAP server, 24-19
 site, 24-10
 SSL, 24-19
config wizard
 command-line options, 10-2

D

database, 2-1
 creating, 2-4
 customizing tablespaces, 2-4
 deleting, 11-5
 deleting tablespaces and datafiles, 11-3
 general requirements, 2-1
 initialization parameters, 2-3

install wizard, 4-7
logging settings, 2-4
Oracle Real Application Clusters, 2-4, 26-1
 installation, 26-3
 post-install steps, 26-3
 pre-install steps, 26-2
patches, 2-2
raw storage, 2-5
repository database, 4-13
requirements, 1-4
disk space, 1-4
 Oracle Beehive Administration Console, 1-8
DMS repository
 updating Oracle Beehive Integration for
 Outlook, 23-4
DMZ, 13-1
 configuring, 13-2
 configuring SSL, 15-5
 high availability, 27-4
 installing, 1-6, 4-14
 installing in silent mode, 9-4
 memory, 1-4
 Oracle Wallet, 13-1
 silent mode, 9-4
 troubleshooting, 13-5

E

e-mail
 configuring e-mail client, 19-6
 configuring with SSL, 19-1
end users
 Oracle Beehive Integration for Outlook
 provisioning, 23-4
 uninstalling, 23-3
 verifying availability, 23-4
enterprise, 4-8

G

Grid Control
 installing provisioning application, 1-5
 provisioning application
 software library, 7-2
 requirements, 1-4
 shiphome binaries location, 1-5

H

- high availability, 27-1
 - DMZ, 27-4
- HTTP
 - configure Apache HTTP server for WebDAV folders, 17-3

I

- install wizard, 4-1
 - checklist, 3-1
 - command-line options, 10-1
 - starting, 1-9
- installation
 - scenarios, 1-4
 - verifying, 9-5
- installing
 - Oracle Beehive Administration Console, 5-1
 - Oracle Beehive Integration for Zimbra, 4-16
 - scenarios
 - against existing database, 1-5
 - DMZ, 1-6, 4-14
 - high availability environment, 1-9, 27-1
 - more than one instance on the same machine, 1-9
 - Oracle Beehive Administration Console, 1-8
 - Oracle Beehive Integration for Zimbra, 1-7
 - provisioning application, 1-5, 4-12, 7-1
 - silent mode, 1-8
 - standard, 1-5, 4-4
 - verifying, 9-3, 9-5
- instance name, 4-9
- inventory location, 4-2, 10-2, 10-3
 - clones, 24-18

J

- J2SE patch cluster, 1-2

K

- Korn shell
 - setting shell limits, 1-3

L

- LDAP
 - LDAP mapping profile, 14-1
 - mapping profile, 14-3
 - retrieving information, 14-13
 - synchronizing with User Directory Service, 14-2
 - user file, 14-1
- LDAP synchronization, 14-1
 - Active Directory considerations, 14-20
 - changing LDAP administrator's password, 14-18
 - cloning, 24-19
 - configuring authentication service, 14-15
 - configuring digest authentication, 14-16
 - configuring SSL, 17-1
 - Directory Replication Group, 14-19

- how it works, 14-2
- LDAP mapping profile, 14-3
- loading users and groups, 14-11
- Oracle Internet Directory considerations, 14-19
- requirements, 14-2
- troubleshooting, 14-21
- troubleshooting replicated LDAP, 24-22

M

- memory, 1-4
 - Oracle Beehive Administration Console, 1-8
- Microsoft Active Directory
 - installing Oracle Beehive Integration for Outlook, 23-3
 - verifying availability for end users, 23-4
 - LDAP mapping profile template, 14-3
- multiple instances, 1-9, 16-2
 - Oracle Real Application Clusters, 26-7

O

- oc4jadmin, 4-10
- Oracle Application Server Certificate Authority, 17-1
- Oracle Beehive Administration Console, 1-8
 - installing, 5-1
 - post-installation, 25-1
- Oracle Beehive Integration for Outlook
 - downgrading, 23-8
 - installing
 - interactively, 23-2
 - silently, 23-2
 - using Microsoft Active Directory, 23-3
 - using remote downloader, 23-1
 - uninstalling for end users, 23-3
 - updating in DMS repository, 23-4
 - XML provisioning file
 - general settings, 23-7
 - registry settings, 23-8
 - sample, 23-10
- Oracle Beehive Integration for Zimbra, 1-7
 - installing, 4-16
- Oracle home, 4-4
- Oracle Internet Directory
 - considerations, 14-19
 - LDAP mapping profile template, 14-3
 - troubleshooting, 14-20
- Oracle Real Application Clusters, 2-4, 26-1, 26-6
 - installation, 26-3
 - multiple instances, 26-7
 - post-install steps, 26-3
 - pre-install steps, 26-2
- Oracle Secure Enterprise Search, 20-1
- oracle software owner user
 - setting shell limits, 1-3
- oracle user
 - setting shell limits, 1-3
- Oracle Wallet, 16-2
 - auto-login, 16-1

- changing password, 16-2
- configuring, 16-1
- configuring TLS, 16-1
- configuring with Oracle Database and Oracle RAC, 15-6
- creating self-signed certificate and importing it, 15-6
- DMZ, 13-1
- multiple instances, 16-2
- ORMIS (ORMI over SSL), 16-2
- oraInventory, 4-2, 10-2, 10-3
- organization, 4-8

P

- port numbers
 - listen on ports less than 1024, 12-1
- ports
 - adding endpoint, 19-3, 19-5
 - BPEL workflows, 12-4
 - creating and configuring endpoint, 19-5
 - customizing ports in cloned instance, 24-18
 - enabling non-SSL and SSL ports, 19-2
 - FTP, 12-5
 - modifying endpoint, 19-2, 19-5
- post-installation, 1-10, 12-1
 - certificate authority, 17-1
 - cloning, 24-1
 - configuring e-mail with SSL, 19-1
 - DMZ, 13-1
 - enabling AJPS, 18-1
 - LDAP synchronization, 14-1
 - listen on ports less than 1024, 12-1
 - Oracle Beehive Administration Console, 25-1
 - Oracle Secure Enterprise Search, 20-1
 - Oracle Wallet, 16-1
 - SSL, 15-1
 - TLS, 16-1
- prerequisites, 1-1
- privileged port numbers, 12-1
- provisioning application, 1-5, 7-1, 8-1
 - installing in silent mode, 9-4
 - introduction, 8-1
 - running, 8-2
 - shiphome binaries location, 1-5
 - silent mode, 9-4

R

- RAC
 - See* Oracle Real Application Clusters
- requirements, 1-1
 - database, 1-4, 2-1
 - general requirements, 2-1
 - initialization parameters, 2-3
 - logging settings, 2-4
 - patches, 2-2
 - hardware, 1-4
 - kernel parameter values, 1-3
 - operating system, 1-1

- Oracle Enterprise Manager Grid Control, 1-4
 - packages
 - SPARC, 1-1
 - patches
 - SPARC, 1-2
- response file, 9-1, 9-2, 9-3, 10-1, 10-2, 10-3
 - creating, 9-2, 9-3

S

- schema password, 4-9
- setting shell limits, 1-3
- shell limits, 1-3
- silent mode, 1-8, 9-1
 - command-line option, 10-3
 - DMZ, 9-4
 - provisioning application, 9-4
 - uninstall, 9-6
 - uninstalling, 9-6
- Single Sign-On, 21-1
- site, 4-9
- SSL, 16-1
 - cloning, 24-19
 - configuring, 15-1
 - Apache HTTP server for WebDAV, 17-3
 - DMZ, 15-5
 - LDAP integration, 17-1
 - real certificates, 15-3
 - test certificates, 15-2
 - LDAP, configure for, 17-1
- swap space, 1-4

T

- tcsh shell
 - setting shell limits, 1-3
- TLS, 16-1
- troubleshooting, A-1
 - DMZ configuration, 13-5
 - LDAP synchronization, 14-21
 - recovering from failed configuration, 9-5
 - replicated LDAP, 24-22
 - synchronizing with Oracle Internet Directory, 14-20

U

- UDS
 - See* User Directory Service
- uninstall, 11-1
 - command-line option, 10-2, 10-3
 - deleting customized tablespaces and datafiles, 11-5
 - deleting tablespaces and datafiles, 11-3
 - manual, 11-3
 - multiple instances, 11-2
 - silent mode, 9-6
- uninstalling, 1-11
 - provisioning application, 7-3
 - silent mode, 9-6
- UNIX users

- setting shell limits, 1-3
- upgrading, 6-1
- User Directory Service, 14-1
- users
 - setting shell limits for UNIX users, 1-3