

Oracle® Beehive

Installation Guide

Release 2 (2.0.1.8) for Linux x86

E16641-07

August 2013

Documentation for administrator that describes how to install, deploy, configure, administer, and maintain Oracle Beehive on Linux x86 operating systems.

Oracle Beehive Installation Guide, Release 2 (2.0.1.8) for Linux x86

E16641-07

Copyright © 2008, 2013, Oracle and/or its affiliates. All rights reserved.

Primary Author: Sujata Nimbalkar

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xix
Audience	xix
Documentation Accessibility	xix
Related Documents	xix
Conventions	xx
 1 What You Should Know Before Installing Oracle Beehive	
Requirements	1-1
Operating System Certification	1-2
Web Browser Certification	1-2
Kernel Version	1-2
Kernel Parameters	1-3
Required Operating System Packages for Linux	1-4
Requirements for 64-Bit Systems	1-6
Install Required Packages	1-6
Use 32-Bit Shell Emulation	1-8
Increasing Shell Limits on Linux	1-8
Database Requirements	1-9
Oracle Enterprise Manager Grid Control Requirements	1-9
Hardware Requirements	1-9
Host Name Resolution	1-10
Installation Scenarios	1-10
Installing Oracle Beehive Against an Existing Database	1-10
Installing Oracle Beehive Integration for Zimbra	1-11
Installing Oracle Beehive Provisioning Application Through Enterprise Manager Grid Control	1-11
Installing Oracle Beehive for Demilitarized Zone (DMZ)	1-11
Installing Oracle Beekeeper	1-13
Requirements and Certifications	1-13
Installing Oracle Beehive in Silent Mode	1-13
Installing Oracle Beehive in High Availability Environment	1-13
Installing More Than One Instance	1-13
Upgrading Oracle Beehive	1-14
Starting Oracle Beehive Install Wizard	1-14
Post-Installation Procedures	1-14

Cloning Application Tiers and Deployments	1-15
Uninstalling Oracle Beehive	1-15

2 Oracle Beehive Database Requirements

General Requirements	2-1
Patches	2-2
Initialization Parameters	2-3
Logging Settings	2-5
Creating a Database	2-5
Using Oracle Real Application Clusters Database	2-5
Customizing Oracle Beehive Tablespaces Layout	2-5
Minimum Tablespace Sizes	2-6
Using Raw Storage	2-6
Upgrading Oracle Database 11g Release 1 to Oracle Database 11g Release 2	2-6
Post-Installation Procedures for Oracle Beehive Databases	2-7

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 Beekeeper	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	4-3
Oracle Beehive Release 2	4-3
Provisioning Application for Oracle Beehive Release 2	4-3
Oracle Beehive Release 2 for DMZ	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
Configure Security Updates	4-7
Database Information	4-8
Template Information	4-9
Enterprise Definition Information	4-10
Existing Enterprise Information	4-12
Schema Password	4-12
Existing Schema Password	4-13
beeadmin Password	4-13
Installation Summary	4-13
Progress Indicator	4-15
Configuration Assistants	4-15

End of Installation	4-15
Oracle Beehive Provisioning Application Sequence of Screens	4-16
Specify Home Location	4-16
Repository Database Administrator Password	4-16
Software Library Configuration.....	4-17
Beehive Software Image Archive.....	4-17
Beekeeper Software Image Archive.....	4-17
Installation Summary	4-17
Installation Progress	4-18
Configuration Assistants.....	4-18
End of Installation	4-18
Oracle Beehive for DMZ Sequence of Screens	4-18
Specify Home Location	4-19
Prerequisite Checks.....	4-19
Installation Summary	4-19
Progress Indicator	4-20
Configuration Assistants.....	4-20
End of Installation	4-20

5 Oracle Beekeeper Installation Help

Product Selection Sequence of Screens	5-1
Welcome	5-1
New Inventory Location	5-2
Select Product Type	5-2
Oracle Beekeeper Sequence of Screens	5-2
Specify Home Location	5-3
Prerequisite Checks.....	5-3
Database Information	5-4
Site Key Information.....	5-4
Installation Summary	5-4
Installation Process	5-5
Configuration Assistants.....	5-5
End of Installation	5-6
Starting the Oracle Beekeeper Uninstall Wizard	5-6
Oracle Beekeeper Uninstall Wizard Sequence of Screens	5-6
Select Oracle Home to Uninstall	5-6
Start Uninstallation	5-6
End of Uninstallation.....	5-7

6 Provisioning Oracle Beehive

Introduction to Oracle Beehive Enterprise Deployment Procedure	6-1
Running Oracle Beehive Enterprise Deployment Procedure	6-2
Deployment Targets	6-3
Adding Targets.....	6-3
Selecting Targets	6-3
Upload	6-4

Configure Security Updates	6-4
Beehive Install Input.....	6-5
Beehive Templates	6-7
Beekeeper Install Input.....	6-8
Exchange/Domino Connector Install Input	6-9
Domino Install Input	6-10
Credentials/Schedule.....	6-10
Summary Page.....	6-11
Uploading New Software Image Archive to Oracle Enterprise Manager Grid Control	6-11
7 Installing Oracle Beehive in Silent Mode (Non-Interactive)	
Installing and Configuring Oracle Beehive in Silent Mode, Running Install Wizard	7-1
Installing and Configuring Oracle Beehive in Silent Mode, Running Install and Config Wizards	7-3
Installing Oracle Beehive Provisioning Application in Silent Mode.....	7-4
Installing Oracle Beehive for DMZ in Silent Mode.....	7-5
Verifying Oracle Beehive Installation	7-5
Recovering from Failed Oracle Beehive Configuration	7-5
Upgrading Oracle Beehive in Silent Mode	7-5
Uninstalling Oracle Beehive in Silent Mode.....	7-6
8 Oracle Beehive Install and Config Wizard Command-Line Options	
Usage.....	8-1
Options	8-1
9 Oracle Beehive Uninstall Wizard	
Starting the Oracle Beehive Install Wizard.....	9-1
Oracle Beehive Uninstall Wizard Sequence of Screens	9-1
Select Oracle Home to Uninstall	9-1
Option to Remove Database Schema	9-2
Start Uninstallation	9-2
End of Uninstallation.....	9-2
Uninstalling Provisioning Application for Oracle Beehive or Oracle Beekeeper.....	9-2
Uninstalling Multiple Instances	9-3
Manually Deleting Oracle Beehive Tablespaces and Datafiles.....	9-3
Manually Deleting Customized Oracle Beehive Customized Tablespaces and Datafiles	9-5
Manually Deleting Existing BEE_CODE Schema.....	9-5
Uninstalling Corrupted Installation.....	9-5
10 Troubleshooting Oracle Beehive Installation	
11 Upgrading Oracle Beehive Overview	
Upgrade Options	11-1
Before Upgrading	11-2
Upgrade Database to Supported Version.....	11-2
Ensure Passwords for Active Code and Data Schemas Are Same.....	11-2

Adjust XmppTimerKeepAliveTime Configuration Parameter	11-2
Analyze Application Tiers	11-3
Rollback Oracle Application Server Critical Patch Update	11-3
Ensure tnsnames.ora File Exists	11-3
Export Configuration Data	11-4
Prepare Oracle Beehive Integration for Zimbra for Upgrade.....	11-4
Upgrading Oracle Beehive Release 1 (1.5.x) with Oracle Beehive Integration for Zimbra Version 1.5.x.....	11-5
Upgrading Previously Upgraded Oracle Beehive Integration for Zimbra Instance.....	11-5
Configure Zero Downtime Upgrade.....	11-6
Configure Oracle Data Pump.....	11-6
Defining Oracle Data Pump Directories for Non-Oracle RAC Deployments	11-7
Defining Oracle Data Pump Directories for Oracle RAC Deployments.....	11-8
Backing up and Deleting Oracle Data Pump Log Files.....	11-8
Disable User Directory Services Synchronization.....	11-8
Shut down All Oracle Beehive Instances.....	11-9
Upgrade Sequence.....	11-9
Upgrading Multiple Oracle Beehive Application Tiers	11-10
Upgrading Oracle Beekeeper Version 1.5.x to Version 2.0.....	11-10
Post-Upgrade Procedures.....	11-11
Running Perl Script post_upgrade_db_actions.pl.....	11-12
Applying Deployment Template after Upgrade	11-14
Gathering Statistics About Oracle Beehive Data and Code Schemas After Upgrading	11-15
Gathering Statistics About Oracle Beehive Data Schema	11-15
Gathering Statistics About Oracle Beehive Code Schema	11-15
Upgrading Voicemail Configuration	11-15
Re-enabling UDS Synchronization	11-16
Upgrading User Directory Services.....	11-17
Changing Permissions of hasbind	11-18
Upgrading Mobile Device Management Service After Oracle Beehive Upgrade	11-18
Upgrading Oracle Beehive by Clone Patching.....	11-19
Creating Batches For Use With add_search_recovery_scope Command	11-19

12 Upgrading Oracle Beehive

Oracle Beehive Upgrade Process Sequence of Screens.....	12-2
Upgrade Existing Home.....	12-2
Specify Home Location to Upgrade	12-2
Running Instances.....	12-3
Configure Security Updates	12-3
Database Information for Upgrade	12-4
Upgrade Summary.....	12-4
Upgrade Process.....	12-5
Configuration Assistants.....	12-5
End of Upgrade	12-5
Oracle Beehive Provisioning Application Upgrade Process Sequence of Screens	12-6
Upgrade Existing Home.....	12-6
Specify Home Location to Upgrade	12-6

Database Information for Upgrade	12-6
Upgrade Summary.....	12-6
Upgrade Process.....	12-7
Configuration Assistants.....	12-7
End of Upgrade	12-7
Oracle Beehive for DMZ Upgrade Process Sequence of Screens	12-7
Upgrade Existing Home.....	12-8
Specify Home Location to Upgrade	12-8
Upgrade Summary.....	12-8
Upgrade Process.....	12-9
Configuration Assistants.....	12-9
End of Upgrade	12-9
 13 Upgrading Oracle Beekeeper	
Oracle Beekeeper Upgrade Process Sequence of Screens	13-1
Upgrade Existing Home.....	13-1
Location to Upgrade	13-1
Prerequisite Checks.....	13-2
Database Information for Upgrade	13-2
Upgrade Summary.....	13-2
Upgrade Process.....	13-3
Configuration Assistants.....	13-3
End of Upgrade	13-3
 14 Oracle Beehive By Clone Patching	
Introduction to Clone Patching	14-1
Types of Clone Patching.....	14-2
Clone Patching	14-4
 15 Oracle Beehive Post-Installation Procedures	
Using Oracle Beehive Command-Line Utility.....	15-2
Performing Post-Installation Procedures on Oracle Beehive Database	15-2
Configuring FTP.....	15-2
Configuring Oracle Beehive to Listen on Ports Less Than 1024	15-2
Changing HTTP Port	15-2
Changing Other Ports.....	15-3
Configuring DMZ Instances	15-5
Integrating and Synchronizing LDAP with Oracle Beehive	15-5
Configuring SSL.....	15-6
Configuring TLS.....	15-6
Configuring SSL for LDAP Integration.....	15-6
Enabling AJPS.....	15-6
Configuring Oracle Beehive E-mail.....	15-6
Configuring XMPP.....	15-7
Configuring Actionable Notifications	15-7
Creating ASK User.....	15-7

Configuring Notifications to Use SMS	15-7
Configuring Oracle Secure Enterprise Search	15-7
Configuring Oracle Application Server Single Sign-On	15-7
Installing Oracle Beehive Extensions for Outlook	15-8
Installing Oracle Beehive Extensions for Explorer	15-8
Installing Oracle Beehive Conferencing Client	15-8
Configuring Oracle Beehive Integration for Zimbra	15-8
Configuring Remote Voice Conferencing Media Server for Oracle Beehive Conferencing ..	15-9
Cloning Oracle Beehive	15-9
Performing Oracle Beekeeper Post-Installation Procedures	15-10
 16 Oracle Beehive Database Post-Installation Procedures	
Modifying Database Connect String, Schema Name, or Credentials	16-1
 17 Configuring FTP	
Enabling Active Mode FTP	17-1
Opening Ports Required by Oracle Beehive for FTP	17-2
Configuring Passive Mode FTP	17-4
Setting other FTP Parameters	17-6
 18 Configuring Oracle Beehive Demilitarized Zone Instances	
Step A: Configuring Oracle Wallet with Oracle Beehive DMZ Instances	18-1
Step B: Configuring Oracle Beehive DMZ Instances	18-2
Configuring Oracle Application Server Single Sign-On with Oracle Beehive DMZ	
Instances	18-5
Troubleshooting DMZ Configuration	18-5
Manually Deleting DMZ Instances	18-5
 19 Configuring SSL	
SSL Checklist	19-1
Configuring SSL with Oracle Beehive	19-2
Configuring SSL with Test Certificates for Oracle Beehive	19-2
Configuring SSL with Self-Signed Certificates During Installation of Oracle Beehive	19-3
Configuring SSL with Self-Signed Certificates After Installation of Oracle Beehive	19-4
Configuring SSL with Oracle Beehive DMZ Instances	19-5
Configuring SSL with Test Certificates After Installation of DMZ Instances	19-5
Configuring SSL with Self-Signed Certificates After Installation of DMZ Instances	19-5
Procedures Related to Configuring SSL	19-6
Creating Self-Signed Certificate and Importing it into Wallet	19-6
Using Oracle Wallet to Create Self-Signed Certificate	19-7
Creating CA-Signed Certificate and Importing it into Wallet	19-7
Installing Non-SSL Oracle Beehive Site	19-8
Installing Oracle Internet Directory in SSL mode	19-9
Configuring the OID in SSL mode	19-11
Appendix	19-12

Configuring Beehive to connect to OID in non-SSL and SSL modes	19-13
Appendix.....	19-16
20 Configuring TLS with Oracle Wallet	
Step 1: Enabling Auto Login Mode for Default Wallet	20-1
Step 2: Configuring Oracle Beehive Instance to Use Oracle Wallet.....	20-2
Step 3: Replacing Test Certificates in Oracle Wallet.....	20-2
Changing Oracle Wallet Password	20-2
Configuring TLS on Multiple Instances.....	20-3
Enabling ORMIS with Password-Protected Oracle Wallet.....	20-3
Disable ORMIS	20-4
Enable ORMIS with Default SSL.....	20-4
Enable ORMIS with Password Protection.....	20-4
21 Configuring SSL for LDAP Integration	
Requirements	21-1
Configure SSL for LDAP Verification	21-2
Configure Apache HTTP Server for WebDAV Folders.....	21-3
22 Configuring E-Mail with SSL	
Securing E-Mail with SSL	22-1
Enabling Both Non-SSL and SSL Ports	22-2
Modifying an Endpoint in SMTP	22-2
Adding an Endpoint to SMTP	22-3
Modifying or Adding an Endpoint in IMAP	22-5
Creating and Configuring an Endpoint.....	22-5
Configuring E-Mail Client	22-6
23 Configuring XMPP	
24 Configuring Oracle Application Server Single Sign-On with Oracle Beehive	
25 Installing Oracle Beehive Extensions for Outlook	
System Requirements.....	25-1
Device Management Service (DMS) Based Installation	25-2
Installing Oracle Beehive Extensions for Outlook Using Remote Downloader	25-2
Installing Oracle Beehive Extensions for Outlook Using Remote Downloader Using Silent Mode	25-3
Specifying Connection Parameters Through Command Line	25-3
Specifying Connection Parameters in Initialization File.....	25-5
Updating and Configuring Oracle Beehive Extensions for Outlook Through DMS	25-6
Updating Oracle Beehive Extensions for Outlook in DMS Repository	25-6
Updating Language Pack for Oracle Beehive Extensions for Outlook	25-7
Applying Configuration Settings from DMS.....	25-8
Customizing Installation Settings	25-10

Creating XML Provisioning File to Customize Installation Settings	25-10
Controlling New Global Address List Settings from DNS.....	25-12
Modifying Registry Settings.....	25-13
Modifying INI File Settings.....	25-13
Enforcing Upgrades.....	25-13
Uninstalling and Downgrading Oracle Beehive Extensions for Outlook (DMS Process) ..	25-14
Uninstalling Oracle Beehive Extensions for Outlook	25-14
Downgrading Oracle Beehive Extensions for Outlook	25-14
Desktop Based Installation (Non-DMS Process)	25-14
Installing Oracle Beehive Extensions for Outlook Using Executable File	25-15
Installing Oracle Beehive Extensions for Outlook in Silent Mode.....	25-15
Configuring Profile for Lockdown Systems Using MSI Package	25-15
Configuring Oracle Beehive Extensions for Outlook Using Configuration Wizard.....	25-17
Configuration Wizard Modes	25-18
Interactive Profile Creation and Configuration Mode	25-18
Interactive PRF File Creation Mode	25-19
Silent Profile Creation Mode	25-20
Upgrading Oracle Beehive Extensions for Outlook Installation	25-21
Updating Language Pack for Oracle Beehive Extensions for Outlook (Non-DMS)	25-21
Uninstalling and Downgrading Oracle Beehive Extensions for Outlook (Non-DMS Process).....	25-22
Uninstalling Oracle Beehive Extensions for Outlook	25-22
Downgrading Oracle Beehive Extensions for Outlook	25-22
Installing Oracle Beehive Extensions for Outlook Using Active Directory	25-22
Creating MSI installation package for Oracle Beehive Extensions for Outlook	25-23
Deploying Oracle Beehive Extensions for Outlook using Active Directory Group Policy	25-23
Verifying that Oracle Beehive Extensions for Outlook Is Available for Installation	25-24
Using Oracle Beehive Extensions for Outlook as Profile Migration Tool.....	25-24
Troubleshooting	25-26

26 Installing Oracle Beehive Extensions for Explorer

System Requirements.....	26-1
Device Management Service (DMS) Based Installation	26-2
Installing Oracle Beehive Extensions for Explorer Using Remote Downloader	26-2
Installing Oracle Beehive Extensions for Explorer Using Remote Downloader Using Silent Mode	26-3
Specifying Connection Parameters Through Command Line	26-3
Specifying Connection Parameters in Initialization File.....	26-4
Update Oracle Beehive Extensions for Explorer Through DMS.....	26-6
Updating Oracle Beehive Extensions for Explorer in DMS Repository.....	26-6
Updating Language Pack for Oracle Beehive Extensions for Explorer	26-7
Uninstalling and Downgrading Oracle Beehive Extensions for Explorer (DMS Process)...	26-8
Uninstalling Oracle Beehive Extensions for Explorer	26-8
Downgrading Oracle Beehive Extensions for Explorer	26-8
Desktop Based Installation (Non-DMS Process)	26-8
Installing Oracle Beehive Extensions for Explorer Using Executable File.....	26-8
Configuring Oracle Beehive Extensions for Explorer Using MSI Package.....	26-9

Custom MSI-based Deployment and Configuration.....	26-9
Upgrading Oracle Beehive Extensions for Explorer Installation	26-10
Updating Language Pack for Oracle Beehive Extensions for Explorer (Non-DMS).....	26-11
Uninstalling and Downgrading Oracle Beehive Extensions for Explorer (Non-DMS Process).....	26-12
Uninstalling Oracle Beehive Extensions for Explorer	26-12
Downgrading Oracle Beehive Extensions for Explorer	26-12
27 Installing Oracle Beehive Conferencing Client	
System Requirements.....	27-1
Device Management Service (DMS) Based Installation	27-3
Installing JavaFX Conferencing Client.....	27-3
Deploying the JavaFX Client	27-3
Installing Oracle Beehive Conferencing Desktop Client Using Remote Downloader	27-3
Updating Oracle Beehive Conferencing in DMS Repository	27-4
Updating Oracle Beehive Conferencing Through DMS.....	27-4
Uninstalling and Downgrading Oracle Beehive Conferencing Client.....	27-5
Uninstalling Oracle Beehive Conferencing Client	27-5
Downgrading Oracle Beehive Conferencing Client	27-5
28 Configuring Oracle Beehive Integration for Zimbra	
Setting Screen Resolution for Oracle Beehive Integration for Zimbra Client.....	28-1
Configuring Oracle Beehive Integration for Zimbra for SSL-Enabled Oracle Beehive Deployment.....	28-1
Changing HTTP Port for Oracle Beehive Integration for Zimbra	28-2
Tuning Oracle Beehive to Improve Performance of Oracle Beehive Integration for Zimbra	28-3
Disabling ORMIS.....	28-4
Disabling SSL	28-4
29 Cloning Oracle Beehive Application Tiers and Sites	
Introduction to Cloning	29-1
Preparing Source Application Tier Instance.....	29-4
Step 1: Verify Requirements	29-4
Step 2: Clear/ Activate Any Pending Configuration Changes to Central Configuration Repository	29-4
Step 3: Shut Down All Processes On the Application Tier	29-5
Step 4: Call beectl clone_preparation Command	29-5
Step 5: Zip Files to Create Clone Image	29-5
Application Tier Cloning.....	29-7
Step 1: Unzip Compressed Oracle Home	29-8
Step 2: Set PERL5LIB Environment Variable	29-9
Step 3: Modify Oracle Home Path	29-9
Step 4: Execute beectl clone_midtier Command.....	29-9
beectl clone_midtier Command	29-9
Step 5: Perform Miscellaneous Operations	29-12
Site Cloning.....	29-12

Step 1: Unzip Compressed Oracle Home	29-12
Step 2: Set PERL5LIB Environment Variable	29-12
Step 3: Modify Oracle Home Path	29-12
Step 4: Execute beectl clone_site Command.....	29-13
beectl clone_site Command	29-13
Step 5: Prevent Services from Target Application Tiers from Referring to External Resources	29-17
Stopping User Directory Service from Referring to LDAP Server	29-17
Stopping Authentication Service from Referring to LDAP Server	29-17
Stopping Virus Scanner Process from Referring to External Virus Scan Engine	29-18
Step 6: Perform Miscellaneous Operations	29-18
OC4J Instance Cloning	29-18
Oracle Beehive Integration for Zimbra Cloning	29-19
Customizing Files or Directories in a Cloned Image	29-19
Customizing Ports in a Cloned Instance	29-20
Cloning Application Tiers and Sites with Ports Less Than 1024	29-20
Oracle Inventory Location Option of Clone Commands on UNIX-Based Systems	29-20
Cloned Application Tiers Are Not Automatically SSL or AJPS Enabled	29-21
Cloning SSL-Enabled Application Tiers	29-21
Cloned Application Tiers and LDAP Synchronization	29-21
Replicating LDAP Server for Cloned Instance	29-22
Testing Replicated LDAP in Cloned Instance	29-24
Troubleshooting Replicated LDAP	29-25
Site Cloning and Multiple Instances	29-25
References to Oracle Application Server Cloning Documentation	29-25
 30 Oracle Beekeeper Post-Installation Procedures	
Configuring Oracle Beekeeper for SSL Access	30-1
Configuring Oracle Beekeeper with Oracle Wallet.....	30-2
Changing Oracle Beekeeper Port	30-3
Changing Oracle Beekeeper RMI Port	30-3
Extending Java Single Sign-On Session Timeout.....	30-3
Configuring Oracle Beekeeper for LDAP-Based Authentication.....	30-4
Reconfiguring Oracle Beekeeper for UDS-Based Authentication.....	30-6
Configuring Virtual Host	30-7
Restarting OC4J	30-8
 31 Enabling AJPS	
Enabling AJPS.....	31-1
Disabling AJPS	31-4
 32 Configuring and Installing Oracle Beehive for Oracle RAC	
Naming Conventions.....	32-1
Pre-Install Steps.....	32-2
Installation	32-2
Post-Install Steps	32-2

Enabling Secure ONS Notification	32-3
Installing Multiple Oracle Beehive Instances with Oracle RAC	32-4
Install All Oracle Beehive Instances, then Apply Post-Install Steps.....	32-4
Install One Oracle Beehive Instance, Apply Post-Install Steps, Install Additional Instances	32-4
Enabling Oracle Beehive Affinity Service	32-4
Troubleshooting	32-6

33 Installing Oracle Beehive in High Availability Environment

Configuring High Availability Environment with Load Balancer.....	33-1
Configuring High Availability Environment with a DMZ Instance	33-3
Configuring High Availability Environment with DMZ Instances and Load Balancer.....	33-4
Configuring SSL Termination at Load Balancer.....	33-6

List of Examples

25-1	Template for Silent Initialization File.....	25-5
25-2	Sample metadata.xml XML Provisioning File	25-9
25-3	Sample XML Provisioning File	25-10
25-4	Template for Migration Initialization File.....	25-24
26-1	Template for Silent Initialization File.....	26-5
26-2	Template for Silent Initialization File.....	26-9

List of Figures

1-1	Oracle Beehive DMZ Instance Integrated in Oracle Beehive System.....	1-12
6-1	Provisioning Advisor Framework Architecture.....	6-2
14-1	Oracle Beehive Deployment Before Clone Patching.....	14-1
14-2	Backing up Application Tier Before Clone Patching	14-2
14-3	Oracle Beehive Deployment After Clone Patching.....	14-2
29-1	Application Tier Cloning	29-2
29-2	Site Cloning.....	29-3
33-1	Multiple Instances in High Availability Environment	33-2
33-2	Multiple DMZ Instances in High Availability Environment	33-5

List of Tables

1-1	Minimum Values and Locations of Kernel Parameters	1-3
1-2	Commands to Display Kernel Parameters	1-3
1-3	Minimum Shell Limits.....	1-8
2-1	Minimum Values of Oracle Database Initialization Parameters.....	2-4
8-1	Install and Config Wizard Command-Line Options	8-1
11-1	Oracle Beehive Upgrade Options for 2.0.1.8.0.....	11-1
14-1	beectl clone_patching Options for Specifying Backed up Oracle home	14-3
14-2	beectl clone_patching Options for Specifying System Model Identifier.....	14-3
16-1	Options for sync_configuration.pl.....	16-2
17-1	Ports Required Open	17-2
17-2	FTP Parameters	17-5
25-1	Oracle Beehive Extensions for Outlook Downloader Options.....	25-4
25-2	Initialization File Parameters	25-5
25-3	Public Properties	25-12
25-4	Installer UI Options	25-15
25-5	MSI Package Public Properties	25-16
25-6	Configuration Wizard Command Line Options	25-17
25-7	Relationship Between Fields in the Profile Options Screen and Settings in the General Section of the PRF File	25-19
25-8	Migration Initialization File Parameters.....	25-25
26-1	Oracle Beehive Extensions for Explorer Downloader Options	26-3
26-2	Initialization File Parameters	26-5
26-3	MSI Package Public Properties	26-9
29-1	beectl clone_preparation Options.....	29-5
29-2	beectl modify_beectl Options.....	29-9
29-3	beectl clone_midtier Options	29-10
29-4	beectl clone_site Options.....	29-14
29-5	beectl clone_oc4j_instance Options	29-19

Preface

The *Oracle Beehive Installation Guide for Linux x86* describes how to install and configure Oracle Beehive.

Audience

The *Oracle Beehive Installation Guide for Linux x86* is directed at any administrator whose task is the installation, deployment, configuration, administration, and maintenance of Oracle Beehive.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Beehive Release 2 (2.0) documentation library:

Administration Guides

- *Oracle® Beehive Administrator's Guide*
- *Oracle® Beehive Administrator's Reference Guide*
- *Oracle® Beekeeper Online Help (Integrated UA)*
- *Oracle® Beehive Integration Guide*

Application Development

- *Oracle® Beehive Application Developer's Guide*
- *Oracle® Beehive Business Views*
- *Oracle® Beehive Java Content Repository Java API Reference*
- *Oracle® Beehive RESTful Web Services API Reference*

- *Oracle® Beehive SOAP Web Services API Reference*

Installation Guides

- *Oracle® Beehive Installation Guide for Microsoft Windows*
- *Oracle® Beehive Installation Guide for Oracle Solaris on SPARC (64-Bit)*
- *Oracle® Beehive Installation Help (Integrated UA)*

Online Helps

- *Oracle® Beehive Central*
- *Oracle® Beehive Webmail*
- *Oracle® Beehive Standards-based Clients*
- *Oracle® Beehive Team Collaboration*
- *Oracle® Beehive Conferencing*
- *Oracle® Beehive Extensions for Explorer Supplemental Help & Release Notes*
- *Oracle® Beehive Extensions for Outlook Supplemental Help & Release Notes*
- *Oracle® Beehive Extensions for Explorer (OBEE) (Integrated UA)*
- *Oracle® Beehive Extensions for Outlook (OBEO) (Integrated UA)*

Mobile Devices

- *Oracle® Beehive Using Windows Mobile Device*
- *Oracle® Beehive Using iPhone or iPad*
- *Oracle® Beehive Using BlackBerry*
- *Oracle® Beehive Registering and Configuring Mobile Devices*

Planning Guides

- *Oracle® Beehive Concepts*
- *Oracle® Beehive Deployment Guide*
- *Oracle® Beehive Licensing Information*

Release Notes

- *Oracle® Beehive Release Notes*

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

The Part I of Installation Guide describes how to install and uninstall Oracle Beehive. To upgrade Oracle Beehive, refer to [Part II, "Oracle Beehive Upgrade Procedures"](#). To configure Oracle Beehive, refer to [Part III, "Oracle Beehive Post-Installation Configuration"](#). To install Oracle Beehive in a high availability environment, refer to [Part IV, "Oracle Beehive High Availability Configuration"](#).

This part contains the following chapters:

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

What You Should Know Before Installing Oracle Beehive

This chapter 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](#)
- [Web Browser Certification](#)
- [Kernel Version](#)
- [Kernel Parameters](#)
- [Required Operating System Packages for Linux](#)
- [Requirements for 64-Bit Systems](#)
- [Increasing Shell Limits on Linux](#)
- [Database Requirements](#)
- [Oracle Enterprise Manager Grid Control Requirements](#)
- [Hardware Requirements](#)
- [Host Name Resolution](#)

The hardware and software requirements included in this installation guide were current at the time this guide was published. However, because new platforms and operating system software versions might be certified after this guide is published, review the certification matrix on the My Oracle Support Web site for the most up-to-date list of certified hardware platforms and operating system versions.

My Oracle Support is available at the following URL:

<http://support.oracle.com/>

You must register online before using My Oracle Support.

Use the following steps to locate the certification information for your platform:

1. After logging in, click the **Certifications** tab.
2. In the Certification Search pane, on the Search tab, select the following information:
 - **Product:** Enter the product name or select from the drop-down list of Products.
 - **Release:** Select the release number of the product.
 - **Platform:** Select the target platform by choosing from the submenu list.

For example, certification matrix for **Oracle Beehive, Release 2.0**, on **Linux x86 Red Hat Enterprise Linux 5**.
3. Click **Search**.

A new window appears with the product certification details.
4. Click the Release Number link to view the certified version of the product on a specific operating system version.

For example, **Oracle Beehive 2.0** is certified with **Oracle Database 11.1.0.7.0** on Operating System **Linux x86 Red Hat Enterprise Linux 5**.
5. Click the Operating System version to view Support Information and Notes.

Note: Contact your Oracle sales representative if you do not have a My Oracle Support account.

Operating System Certification

Oracle Beehive supports the following Linux (x86 32-bit) operating system certifications:

- Oracle Enterprise Linux 4
- Oracle Enterprise Linux 5
- Red Hat Enterprise Linux 4 Update 3 and later
- Red Hat Enterprise Linux 5
- SuSE Linux Enterprise 10.0

Web Browser Certification

- Mozilla Firefox 3.5 and 3.6
- Google Chrome 10 and 11

Kernel Version

The system must be running the following kernel version (or a later version):

- Oracle Enterprise Linux 4 and 5: 2.6.9-34.0.1.0.11.EL

- Red Hat Enterprises Linux 4 and 5: 2.6.9-29.EL
- SuSE Linux Enterprise 10.0: 2.6.16.21-0.8

Note: Oracle Beehive also supports both SMP (symmetric multiprocessing) and hugemem kernels. For example, Oracle Beehive also supports Red Hat Enterprises Linux 4 and 5 kernel versions 2.6.9-29.ELsmp and 2.6.9-29.ELhugemem.

Kernel Parameters

Verify that the kernel parameters shown in the following table are set to values greater than or equal to the minimum value shown. If the current value for any parameter is higher than the value listed in this table, then do not change the value of that parameter. The procedure following the table describes how to verify and set the values.

Note: You require root privileges to perform the steps described in this section.

The kernel parameter and shell limit values shown in the following section are minimum values only. For production database systems, Oracle recommends that you tune these values to optimize the performance of the system. Refer to the operating system documentation for more information about tuning kernel parameters.

Table 1–1 Minimum Values and Locations of Kernel Parameters

Parameter	Minimum Value	File
ip_local_port_range	Minimum: 22000 Maximum: 65000	/proc/sys/net/ipv4/ip_local_port_range This file defines the local port range that is used by TCP and UDP traffic to choose the local port.

To display the current value specified for these kernel parameters, and to change them if necessary, use the following steps:

1. Enter the commands shown in the following table to display the current values of the kernel parameters; make a note of these values and identify any values that you must change:

Table 1–2 Commands to Display Kernel Parameters

Parameter	Command
ip_local_port_range	# /sbin/sysctl -a grep ip_local_port_range This command displays a range of port numbers.

2. If the value of any kernel parameter is different from the minimum value, then complete the following procedure:
 1. Using any text editor, create or edit the /etc/sysctl.conf file, and add or edit lines similar to the following:

```
net.ipv4.ip_local_port_range = 22000 65000
```

By specifying the values in the `/etc/sysctl.conf` file, they persist when you restart the system. However, on SuSE Linux Enterprise Server systems, enter the following command to ensure that the system reads the `/etc/sysctl.conf` file when it restarts:

```
# /sbin/chkconfig boot.sysctl on
```

2. Enter the following command to change the current values of the kernel parameters:

```
# /sbin/sysctl -p
```

Review the output from this command to verify that the values are correct. If the values are incorrect, edit the `/etc/sysctl.conf` file, then enter this command again.

3. Enter the command `/sbin/sysctl -a` to confirm that the values are set correctly.
4. On SuSE Linux systems only, enter the following command to cause the system to read the `/etc/sysctl.conf` file when it restarts:

```
# /sbin/chkconfig boot.sysctl on
```

5. On SuSE Linux systems only, you must enter the GID of the `oinstall` group as the value for the parameter `/proc/sys/vm/hugetlb_shm_group`. Doing this grants members of `oinstall` a group permission to create shared memory segments.

For example, where the `oinstall` group GID is 501:

```
# echo 501 > /proc/sys/vm/hugetlb_shm_group
```

After running this command, use `vi` to add the following text to `/etc/sysctl.conf`, and enable the `boot.sysctl` script to run on system restart:

```
vm.hugetlb_shm_group=501
```

Note: Only one group can be defined as the `vm.hugetlb_shm_group`.

6. After updating the values of kernel parameters in the `/etc/sysctl.conf` file, either restart the computer, or run the command `sysctl -p` to make the changes in the `/etc/sysctl.conf` file available in the active kernel memory.

Required Operating System Packages for Linux

Oracle Beehive requires the following packages. (This list also indicates the minimum version required for each package.)

Note: The versions of these packages may differ between different Linux distributions.

To determine whether a required package is installed, use the `rpm` command:

```
rpm -q <package name>
```

If a package is not installed, install it from the Linux distribution media or download the required package from your Linux vendor's web site.

- **Oracle Enterprise Linux 4 and Red Hat Enterprise Linux 4**

- gcc-3.4.5-2 (GNU Compiler Collection)
- gcc-c++-3.4.5-2 (C++ support for GCC)
- openmotif21-2.1.30-11.RHEL4.4 (Open Motif run time libraries and executable files)
- setarch-1.6-1 (Personality setter)
- pdksh-5.2.14-30.3 (Public domain Korn shell)
- sysstat-5.0.5-7 (System monitoring commands sar and iostat)
- gnome-libs-1:1.4.1.2.90-44.1 (GNOME basic libraries)
- libstdc++-3.4.5-2 (GNU Standard C++ Library)
- libstdc++-devel-3.4.5-2 (Header files and libraries for C++ development)
- compat-libstdc++-296-2.96-132.7.2 (Compatibility 2.96-Red Hat standard C++ libraries)
- compat-db-4.1.25-9 (Berkeley DB database library for Red Hat Linux 7.x compatibility)
- control-center-2.8.0-12rhel4.2 (GNOME Control Center)
- glibc-common-2.3.4-2.9 (Common binaries and locale data for glibc)
- binutils-2.15.92.0.2-18 (GNU Binary Utility Development Utilities)
- make-1:3.80-5 (GNU make)
- xscreensaver-4.18-5.rhel4.10 (X Window System screen saver and locker)
- xorg-x11-deprecated-libs-6.8.2-1.EL.13.25 (Deprecated X Window System shared libraries)

- **Oracle Enterprise Linux 5 and Red Hat Enterprise Linux 5**

- binutils-2.16.91.0.5-23.4.i386.rpm
- gcc-4.1.0-28.4.i386.rpm
- gcc-c++-4.1.0-28.4.i386.rpm
- glibc-devel-2.4-31.2.i386.rpm
- libgcc-4.1.0-28.4.i386.rpm
- libstdc++-devel-4.1.0-28.4.i386.rpm
- libstdc++-4.1.0-28.4.i386.rpm

- make-3.80-202.2.i386.rpm
- **SuSE Linux Enterprise 10.0**
 - ksh-93r-12.9 (Korn shell)
 - sysstat-5.0.1-35.1 (System monitoring commands sar and iostat)
 - openmotif21-libs-2.1.30MLI4-143.2 (Open Motif run time libraries and executable files)
 - libstdc++-4.1.0-28.4 (GNU Standard C++ Library)
 - make-3.80-202.1 (GNU make)
 - binutils-2.16.91.0.5-23.4 (GNU Binary Utility Development Utilities)
 - gcc-4.1.0-28.4 (GNU Compiler Collection)
 - gcc-c++-4.1.0-28.4 (C++ support for GCC)
 - libstdc++-devel-4.1.0-28.4 (GNU Standard C++ Library)
 - db1-1.85-101.2 (Berkeley DB database library)

Requirements for 64-Bit Systems

To install and deploy Oracle Beehive on 64-bit systems, perform the following:

- [Install Required Packages](#)
- [Use 32-Bit Shell Emulation](#)

Install Required Packages

Ensure that the following packages are installed on your system. (This list also indicates the minimum version required for each package.)

Note: On Red Hat Enterprise Linux systems, to install 32-bit packages on a 64-bit system, you may need to use the `--force` option and the `--nodeps` option of the `rpm` utility.

- **All supported versions of Linux:**
 - gdbm-1.8.0-26.2.1-i386.rpm (32-bit)
 - glibc-devel-2.3.4-2.19.i686.rpm (32-bit)
 - glib-devel-1.2.10-15-i386.rpm (32-bit)
 - glib-1.2.10-15-i386.rpm (32-bit)
- **Oracle Enterprise Linux 5 and Red Hat Enterprise Linux 5**
 - binutils-2.16.91.0.5-23.4.x86_64.rpm
 - gcc-4.1.0-28.4.x86_64.rpm
 - gcc-c++-4.1.0-28.4.x86_64.rpm
 - glibc-devel-2.4-31.2.i386
 - glibc-devel-2.4-31.2.x86_64
 - libgcc-4.1.0-28.4.el5.i386
 - libgcc-4.1.0-28.4.el5.x86_64

- libstdc++-devel-4.1.0-28.4.i386.rpm
- libstdc++-devel-4.1.0-28.4.x86_64.rpm
- libstdc++-4.1.0-28.4.i386.rpm
- libstdc++-4.1.0-28.4.x86_64.rpm
- make-3.80-202.2.x86_64.rpm

Note: To install Oracle Beehive with Simplified Chinese and Traditional Chinese language on update level 2 or lower, install the package fonts-chinese-3.02-12.el5.noarch.rpm. Obtain this package from the following location:

http://oss.oracle.com/projects/compat-oracle/files/Enterprise_Linux/

■ **SuSE Linux Enterprise 10.0:**

- glibc-2.3.3-98.28.x86_64.rpm
- glibc-32bit-9-200407011233.x86_64.rpm
- glibc-devel-32bit-9-200407011229.x86_64.rpm
- gcc-3.3.3-43.24.x86_64.rpm
- gcc-3.2.3-20.x86_64.rpm
- gcc-c++-3.3.3-43.24.x86_64.rpm
- libgcc-3.2.3-20.i386.rpm (32-bit)
- libstdc++-3.3.3-43.24.x86_64.rpm
- libstdc++-devel-3.3.3-43.24.x86_64.rpm
- libstdc++-devel-32bit-9-200407011229.x86_64.rpm
- gnome-libs-1.4.1.7-671.1.x86_64.rpm
- gnome-libs-devel-1.4.1.7-671.1.x86_64.rpm
- gnome-libs-32bit-9-200407011229.x86_64.rpm
- binutils-2.15.90.0.1.1-32.5.x86_64.rpm
- binutils-32bit-9-200407011229.x86_64.rpm
- compat-32bit-9-200407011229.x86_64.rpm
- compat-sles8-1.3-93.3.x86_64.rpm
- compat-2004.7.1-1.2.x86_64.rpm
- pdksh-5.2.14-780.1.x86_64.rpm
- make-3.80-184.1.x86_64.rpm
- sysstat-5.0.1-35.1.x86_64.rpm
- perl-32bit-9-200407011229.x86_64.rpm
- libaio-devel-32bit-9-200407011229.x86_64.rpm
- XFree86-devel-32bit-9-200407011229.x86_64.rpm
- linux32-1.0-341.1.x86_64.rpm

- db-32bit-9-200407011229.x86_64.rpm
- db1-32bit-9-200407011229.x86_64.rpm
- compat-32bit-9-200407011229.x86_64.rpm
- db1-1.85-119.i586.rpm
- compat-gdbm-1.8.0-7.i586.rpm

Use 32-Bit Shell Emulation

If you plan to deploy Oracle Beehive for Linux on a 64-bit system, you must use 32-bit shell emulation. To do this, use the following command before running the Install Wizard or any other Oracle Beehive commands:

```
linux32 bash
```

Increasing Shell Limits on Linux

Note: In order to install Oracle Beehive, the maximum number of open file descriptors must be at least 4096.

Although the steps in this section increase the value of the maximum number of open file descriptors, ensure that the kernel parameter `fs.file-max` is set to a value equal or higher than the value of `hard nfile` in the `/etc/security/limits.conf` file.

To improve the performance of Oracle Beehive on Linux, increase the following shell limits for the user installing Oracle Beehive:

Table 1–3 Minimum Shell Limits

Shell Limit	Item in limits.conf	Hard Limit
Maximum number of open file descriptors	nfile	65536
Maximum number of processes available to a single user	nproc	16384

Follow these steps to increase the shell limits. These steps assume you are using a user named `oracle` to install Oracle Beehive:

1. Add the following lines to the `/etc/security/limits.conf` file:

```
oracle      soft    nproc    2047
oracle      hard    nproc    16384
oracle      soft    nfile    1024
oracle      hard    nfile    65536
```

2. Add or edit the following line in the `/etc/pam.d/login` file, if it does not already exist:

```
session required /lib/security/pam_limits.so
```

3. Depending on the `oracle` user's default shell, make the following changes to the default shell start-up file:

- For the Bourne, Bash, or Korn shell, add the following lines to the `/etc/profile` file (or the file on SuSE Linux systems, `/etc/profile.local`):

```
if [ $USER = "oracle" ]; then
```

```

        if [ $SHELL = "/bin/ksh" ]; then
            ulimit -p 16384
            ulimit -n 65536
        else
            ulimit -u 16384 -n 65536
        fi
    fi

```

- For the C shell (csh or tcsh), add the following lines to the `/etc/csh.login` file (or the file on SuSE Linux systems, `/etc/csh.login.local`):

```

if ( $USER == "oracle" ) then
    limit maxproc 16384
    limit descriptors 65536
endif

```

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.

Oracle Enterprise Manager Grid Control Requirements

If you want to install the Oracle Beehive Provisioning Application, which enables you to install Oracle Beehive from Oracle Enterprise Manager Grid Control, you must have Oracle Enterprise Manager 10g Release 5 Grid Control (10.2.0.5) or later.

Hardware Requirements

The following are the minimum hardware requirements for Oracle Beehive:

- Disk Space: 2.6 GB

Note: Installing Oracle Beehive on a Network File System (NFS) file system is currently not supported.

- Swap Space: Oracle Beehive requires at least 2,048 MB available swap space.
- Memory:
 - Oracle Beehive (refer to ["Template Information"](#) for more information about these templates):
 - * Basic Server Template: 2 GB
 - * Basic Server and Client Template: 3 GB
 - * Client Only Template: 2 GB
 - Oracle Beehive for DMZ: 1 GB

Note: 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.

Host Name Resolution

Before installing Oracle Beehive, configure host name resolution and verify that your host name or host names resolve.

1. Verify that the host name has been set by using the `hostname` command:

```
hostname
```

The output of this command should be similar to the following:

```
myhost.mycomputer.com
```

2. Verify that the `/etc/hosts` file contains the fully qualified host name by using the following command:

```
cat /etc/hosts | grep 'eval hostname'
```

The output of this command should contain an entry for the fully qualified host name and for localhost. For example:

192.168.100.16	myhost.mycompany.com	myhost
127.0.0.1	localhost	localhost.localdomain

If the `/etc/hosts` file does not contain the fully qualified host name, then edit the file and make the required changes in it.

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 Through Enterprise Manager Grid Control](#)
- [Installing Oracle Beehive for Demilitarized Zone \(DMZ\)](#)
- [Installing Oracle Beekeeper](#)
- [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 Integration for Zimbra

Oracle Beehive Integration for Zimbra is a Web-based client for Oracle Beehive. To install this product, select the Basic Server and Client template when installing Oracle Beehive. This will install Oracle Beehive and Oracle Beehive Integration for Zimbra in the same location. Alternatively, if you have already installed at least one Oracle Beehive instance, you may select the Client Only template, which enables you to install Oracle Beehive Integration for Zimbra in a different location than your Oracle Beehive instance.

For more information, refer to ["Template Information"](#) in ["Oracle Beehive \(Standard Installation\) Sequence of Screens"](#).

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

Installing Oracle Beehive Provisioning Application Through Enterprise Manager Grid Control

Oracle Enterprise Manager 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 Enterprise Manager Grid Control gives you a broad set of administration, configuration management, provisioning, end-to-end monitoring, and security capabilities.

Oracle Beehive Provisioning Application enables you to install and configure Oracle Beehive products in the robust environment of Oracle Enterprise Manager Grid Control.

Oracle Beehive Provisioning Application enables you to upload installable versions of Oracle Beehive products into the central software library of Oracle Enterprise Manager Grid Control. It also creates the Oracle Beehive Enterprise Deployment Procedure with which you can install and configure Oracle Beehive products on any host through the Deployment Procedure Manager of Oracle Enterprise Manager Grid Control.

Refer to ["Oracle Beehive Provisioning Application Sequence of Screens"](#) and ["Provisioning Oracle Beehive"](#) for more information.

Note: If you have any additional management service instances connected to the Management Repository, you must shut down all additional management service instances before installing Oracle Beehive Provisioning Application.

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.

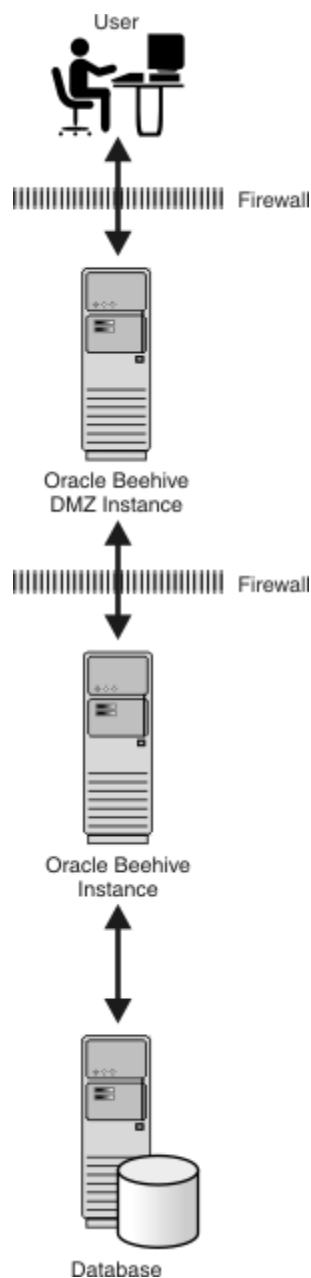
An Oracle Beehive DMZ configuration consists of one or more Oracle Beehive DMZ instances that are isolated from the Internet by a firewall on one side, and from one or more Oracle Beehive non-DMZ instances by a firewall on the other side. The DMZ instances are viewed as semi-secure. They are protected 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 Beekeeper

Oracle Beekeeper 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, refer to ["Oracle Beekeeper Installation Help"](#).

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

Requirements and Certifications

The following are the minimum hardware requirements for Oracle Beekeeper:

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

Installing Oracle Beehive in Silent Mode

Silent mode enables 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.

Note: If your Oracle Beehive deployment will be using WebDAV, your load balancer must be RFC 2518 (*HTTP Extensions for Distributed Authoring -- WEBDAV*) compliant.

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 computer 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

Refer to ["Upgrading Oracle Beehive Overview"](#) for steps to perform before upgrading your Oracle Beehive deployment to Oracle Beehive Release 2 (2.0), the order in which you should upgrade Oracle Beehive products, and other procedures to perform after upgrading.

Starting Oracle Beehive Install Wizard

All installation scenarios use the Oracle Beehive Install Wizard.

Note: Before installing Oracle Beehive, check if the file `tnsnames.ora` exists in the directory `/etc` or `/var/opt/oracle`. If so, either move or rename the `tnsnames.ora` file prior to installing Oracle Beehive.

After installing and configuring Oracle Beehive, you may move or rename the file to its original location or name.

Run `runInstaller` from a user account other than root to start the installation and configuration of Oracle Beehive.

Note: For SuSE Linux Enterprise, set the environment variable `LD_LIBRARY_PATH` to `<Oracle Database home>/lib` and start the database before running the Install Wizard.

Note: Oracle Beekeeper 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 Beekeeper.

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](#)
- [Performing Post-Installation Procedures on Oracle Beehive Database](#)
- [Configuring FTP](#)
- [Configuring Oracle Beehive to Listen on Ports Less Than 1024](#)
- [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 XMPP](#)
- [Configuring Actionable Notifications](#)
- [Creating ASK User](#)
- [Configuring Notifications to Use SMS](#)
- [Configuring Oracle Secure Enterprise Search](#)
- [Configuring Oracle Application Server Single Sign-On](#)
- [Installing Oracle Beehive Extensions for Outlook](#)
- [Installing Oracle Beehive Extensions for Explorer](#)
- [Installing Oracle Beehive Conferencing Client](#)
- [Configuring Oracle Beehive Integration for Zimbra](#)
- [Configuring Remote Voice Conferencing Media Server for Oracle Beehive Conferencing](#)
- [Cloning Oracle Beehive](#)
- [Performing Oracle Beekeeper 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 2 (2.0)
- Oracle Beehive Provisioning Application
- Oracle Beehive for DMZ

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 \(Non-Interactive\)](#)".

Note: The Oracle Beehive Uninstall Wizard cannot remove the Oracle Beehive schema of a site-cloned application tier. Refer to "[Manually Deleting Oracle Beehive Tablespaces and Datafiles](#)" in "[Oracle Beehive Uninstall Wizard](#)" to manually remove the schema.

Oracle Beehive Database Requirements

This chapter 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](#)
- [Upgrading Oracle Database 11g Release 1 to Oracle Database 11g Release 2](#)
- [Post-Installation Procedures for Oracle Beehive Databases](#)

General Requirements

Oracle Beehive requires an existing database with the following characteristics:

- Supports Oracle Database 11g Release 1 and 2 (11.1.0.7, 11.2.0.1, 11.2.0.2, and 11.2.0.3) Enterprise Edition, 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 Multimedia in Oracle Database 11g)
 - Oracle Text
 - Partitioning
- Contains Rules Manager and Expression Filter (Rules Manager and Expression Filter are installed automatically with Oracle Database 11g Enterprise Edition.)

Patches

Microsoft Windows

For Microsoft Windows (32-bit and 64-bit versions), install the latest patch bundle for your database.

Note: Refer to latest Patchset Readme for information on: *Patch Required for Upgraded Beehive Instances on Oracle Database 11.2.*

The latest Patchset Readme is available on the Beehive Documentation Library at: <http://www.oracle.com/pls/bee2/homepage>

Operating Systems Other Than Microsoft Windows

Install the following patches for Oracle Database 11g Release 1 (11.1.0.7):

- 9548753: MERGE REQUEST ON TOP OF 11.1.0.7.0 FOR BUGS 9128529 8499043
- 8537211: MERGE LABEL REQUEST ON TOP OF 11.1.0.7 FOR BUGS 6625210 7694979 7378322
- 8531282: MERGE LABEL REQUEST ON TOP OF 11.1.0.7 FOR BUGS 8352309 7586451
- 8499600: ORA-01555 ERRORS ON SECUREFILES READS EVEN UNDER LOW LOADS/NO UPDATE ACTIVITY
- 8487273: MERGE LABEL REQUEST ON TOP OF 11.1.0.7 FOR BUGS 8214576 7258928
- 8221425: STBH:HP:11G - ORA-04031: UNABLE TO ALLOCATE 912 BYTES OF SHARED MEMORY
- 7707103: RECOMPILATION OF LIBRARY UNIT GIVES END-OF-FILE COMMUNICATION CHANNEL
- 7690421: INVALID TZ VALUE IN TIMEZONE COLUMN PRODUCES ORA-01875
- 7685452: 10.2.0.5.0 : HP-PA : ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [4449], [],
- 7643188: SQL VERSION COUNT REPORTED IN AWR DOES NOT MATCH SQLSTATS FOR 1 SQL ID
- 7500792: HIGH GETS ON "KOKC DESCRIPTOR ALLOCATION LATCH" WITH MORE THAN 400 USERS
- 7313961: MISSING LSFEND IN DRUERE() CALLS ADDED FOR BUG 6499258
- 7311226: PARTITIONED TABLE AND FILTER BY OPTIMIZE_INDEX NOT PICKING UP WHERE IT LEFT OFF
- 7273988: TST&PERF:QUERY PLAN IS VERY BAD IN RDBMS_MAIN_LINUX_080623
- 7156912: ORACLE TEXT INDEXING HANGS ON A HTML DOCUMENT WITH LONG COMMENTS
- 7047984: ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [K]BCLOSE:MAS], [3], [1]...
- 6977167: ORA-04092: CANNOT ROLLBACK IN A TRIGGER RECEIVED REQUIRING INSTANCE BOUNCE

- 6782437: QUERY WITH MULTIPLE SUBQUERIES PERFORMING POORLY AFTER 10G UPGRADE
- 6750049: LIBNNZ10.SO DOPEN FAILED WHEN TRYING TO START ONS (NODEAPPS)
- 6698219: DBMS_LINK ALLOWS PATH > 1024 CAUSING CORRUPT REPOSITORY
- 6355412: JOB CONSUMES 100% CPU FOR EXTENDED PERIODS
- 6263237: TCH11G: ORA-7445 WHILE RUNNING ADPATCH
- 6083201: MERGE STATEMENT WITH XML TYPE FAILS WITH ORA-7445 [FCLOSE()+29]

Install the following patches for Oracle Database 11g Release 1 (11.1.0.7.2):

Note: Install patches in the following order:

1. Oracle Database 11g Release 1 (11.1.0.7) patches as specified in this document
 2. Oracle Database 11g Release 1 (11.1.0.7.2) Patch Set Update; for more information about Patch Set Updates, refer to Oracle My Support Note 854428.1
 3. Oracle Database 11g Release 1 (11.1.0.7.2) as specified in the following list
-

- 9348916: MERGE REQUEST ON TOP OF 11.1.0.7.2 FOR BUGS 6856345 7284151 8465729 9010222
- 7047984: ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [KJBCLOSE:MAS], [3], [1]...
- 6782437: QUERY WITH MULTIPLE SUBQUERIES PERFORMING POORLY AFTER 10G UPGRADE
- 6625210: ORA-22370: INCORRECT USAGE OF METHOD ANYDATA INSERT

Install the following patches for Oracle Database 11g Release 2 (11.2.0.1):

- 9275876: PEOPLESFT QUERY FAILED WITH CORE DUMP ORA 7445 QERIXGETKEY
- 9033671: REGR: ORA-07445: EXCEPTION ENCOUNTERED: CORE DUMP [KKQFPPDRV1()+66] [SIGSEGV]
- 9010222: APPS ST 11G ORA-00600 [KKSFBC-REPARSE-INFINITE-LOOP]
- 8772524: ORA-38500: EXPRESSION IS INVALID IF WE USE THE IN OPERATOR

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	11.1.0.0.0 Note: Do not modify this parameter if it is already set to a higher value. Ensure that the value of this parameter is in the form <i>x.x.x.x.x</i> (for example, 11.1.0.7.0). If this format isn't used, you may encounter a validation error stating that the compatible parameter is not of the proper minimum version.
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 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. If any of the db_nk_cache_size parameters are set, then Oracle Beehive will use the lowest value that is equal or greater than 8192. Refer to "Specifying Database Block Sizes" in <i>Oracle Database Administrator's Guide</i> for more information.
java_pool_size	50M (52428800)
job_queue_processes	10
processes	150
sga_max_size	1G
sga_target	500M
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 11.1.0.7.0, use the following command:

```
SQL> ALTER SYSTEM SET COMPATIBLE = '11.1.0.7.0' SCOPE = SPFILE;
```

Note: After setting initialization parameters, restart the database.

Logging Settings

If your Oracle Beehive deployment is a production system, it is strongly recommended that you set the archive log mode of the database to ARCHIVELOG, which enables the archiving of the redo logs. 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 "Creating a Database Using DBCA" in Chapter 2, "Installing Oracle Database and Creating a Database" in *Oracle Database 2 Day DBA*.

Tip: When creating a database for Oracle Beehive with DBCA, you may choose any template. With DBCA, 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 (Oracle 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.

Minimum Tablespace Sizes

By default, Oracle Beehive tablespaces are created with `autoextend` enabled. If you prefer to manage the size of the Oracle Beehive tablespaces yourself, the total size of all Oracle Beehive tablespaces should be at least 15 GB.

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.

Upgrading Oracle Database 11g Release 1 to Oracle Database 11g Release 2

If you upgrade Oracle Database 11g Release 1 (11.1.0.7) to Oracle Database 11g Release 2 (11.2) after installing Oracle Beehive, follow these steps:

1. Shut down Oracle Beehive:

```
beectl stop --all
```

2. Run the following SQL scripts as the `BEE_CODE` user in the specified order:

- a. `<Oracle Beehive home>/beehive/db/framework/ocs_stop_jobs.sql`

- b. `<Oracle Beehive home>/db/dbclone/drop_expression_indexes.sql`

- c. `<Oracle Beehive home>/db/dbclone/create_expression_indexes.sql`

d. `<Oracle Beehive home>/db/framework/ocs_enable_jobs.sql`

3. Start Oracle Beehive:

```
beectl start --all
```

Post-Installation Procedures for Oracle Beehive Databases

After you have successfully installed Oracle Beehive, refer to "[Oracle Beehive Database Post-Installation Procedures](#)" for tuning recommendations to perform on your Oracle Beehive database.

Oracle Beehive Install Wizard Checklist

This chapter lists the information the Oracle Beehive Install Wizard requests during the installation process for the following products:

- [Oracle Beehive \(Standard Installation\)](#)
- [Oracle Beehive Provisioning Application](#)
- [Oracle Beehive for DMZ](#)
- [Oracle Beekeeper](#)

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 where you want to install Oracle Beehive.
- [Prerequisite Checks](#): Refer to "Requirements" in "What You Should Know Before Installing Oracle Beehive" for Oracle Beehive software and hardware requirements.
- [Database Information](#): Refer to "Oracle Beehive Database Requirements" for database requirements.
 - [Host and Port](#)
 - [Service Name](#)
 - [Admin User Name](#)
 - [Admin User Password](#)
- [Template Information](#)
 - [Select Template](#): Templates include options to install Oracle Beehive Integration for Zimbra.
 - [Memory to Allocate](#)
- [Enterprise Definition Information](#) or [Existing Enterprise Information](#)
 - [Enterprise](#)
 - [Organization](#)
 - [Site](#)

- [Site Key](#)
- [Instance](#)
- [Schema Password](#) or [Existing Schema Password](#)
- [beeadmin Password](#)

Oracle Beehive Provisioning Application

- [New Inventory Location](#): Oracle inventory directory path.
The Oracle Beehive Install Wizard will ask you for the Oracle inventory directory path if this is the first Oracle product you are installing on your system.
- [Specify Home Location](#): Full path of the location where you want to install Oracle Beehive Provisioning Application.
- [Repository Database Administrator Password](#)
- [Software Library Configuration](#): Full path of a directory that will contain software image archives of Oracle Beehive products in binary format.
- [Beehive Software Image Archive](#)
- [Beekeeper Software Image Archive](#)

Oracle Beehive for DMZ

- [New Inventory Location](#): Oracle inventory directory path.
The Oracle Beehive Install Wizard will ask you for the Oracle inventory directory path if this is the first Oracle product you are installing on your system.
- [Specify Home Location](#): Full path of the location where you want to install Oracle Beehive for DMZ.
- [Prerequisite Checks](#): Refer to "[Requirements](#)" in "[What You Should Know Before Installing Oracle Beehive](#)" for Oracle Beehive software and hardware requirements.

Oracle Beekeeper

- [New Inventory Location](#): Oracle inventory directory path.
The Oracle Beehive Install Wizard will ask you for the Oracle inventory directory path if this is the first Oracle product you are installing on your system.
- [Specify Home Location](#): Full path of the location where you want to install Oracle Beekeeper.
- [Prerequisite Checks](#): Refer to "[Requirements and Certifications](#)" in "[What You Should Know Before Installing Oracle Beehive](#)" for Oracle Beekeeper requirements and certifications.
- [Database Information](#): Refer to "[Oracle Beehive Database Requirements](#)" for database requirements.
 - [Host and Port](#)
 - [Service Name](#)
 - [Schema User Password](#)

- [Site Key Information](#)

Oracle Beehive Install Wizard

This chapter describes 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](#)

Note: To install Oracle Beehive, run `setup.exe` (Microsoft Windows) or `runInstaller` (UNIX-based operating systems) from the Oracle Beekeeper installation media.

Product Selection Sequence of Screens

The following screens appear before 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](#)

Welcome

The Welcome screen is the first screen of the Oracle Beehive Install Wizard. It will guide you through the installation and configuration of Oracle Beehive Release 2.

Preview Mode

Selecting the check box **Preview Mode** enables you to run the Install Wizard in preview mode. In this mode, the Install Wizard does not validate any of your input and disables the **Install** and **Uninstall** buttons. Consequently, the preview mode enables you to view all the screens of the Install Wizard without installing any Oracle Beehive products.

Install Wizard Buttons

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 "[Installing Oracle Beehive in Silent Mode \(Non-Interactive\)](#)" 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 2 is the first Oracle product you are installing on a computer, then this screen is displayed.

Inventory Directory Path

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

Note: Note the following 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 2.
- 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`.

Operating System Group

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 2](#) option.
- Plug-in installation by using the [Provisioning Application for Oracle Beehive Release 2](#) option.
- Installation for a demilitarized zone by using the [Oracle Beehive Release 2 for DMZ](#) option.

The following sequence of screens depend on the option that you select in the above option.

Oracle Beehive Release 2

Use this option to perform a complete installation of Oracle Beehive Release 2 on the specified host.

Note: One of the prerequisites for selection the Oracle Beehive Release 2 option is that you must have an existing instance of Oracle Database. This database is not installed by this procedure.

Refer to "[Oracle Beehive Database Requirements](#)" for database requirements for Oracle Beehive Release 2 (2.0).

See "[Oracle Beehive \(Standard Installation\) Sequence of Screens](#)" to continue installing Oracle Beehive Release 2.

Provisioning Application for Oracle Beehive Release 2

Use this option to install the Provisioning Application. This will allow you to install and configure Oracle Beehive Release 2 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 5 (10.2.0.5) or later.

See "[Oracle Beehive Provisioning Application Sequence of Screens](#)" to continue installing Oracle Beehive Release 2 Provisioning Application.

Oracle Beehive Release 2 for DMZ

Use this option to install Oracle Beehive Release 2 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 2 for DMZ.

Oracle Beehive (Standard Installation) Sequence of Screens

The following screens appear during the installation of Oracle Beehive Release 2 (standard installation):

- [Specify Home Location](#)
- [Prerequisite Checks](#)
- [Select Installation Type](#)
- [Configure Security Updates](#)
- [Database Information](#)
- [Template 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 2** 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 2](#)". Refer to "[What You Should Know Before Installing Oracle Beehive](#)" for information on prerequisites for installing Oracle Beehive Release 2.

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 2) 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**.

Note: 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 2 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.

Configure Security Updates

Specify your My Oracle Support account details so that Oracle can notify you of any critical security updates.

Periodically, Security Updates automatically gathers configuration information of your installed Oracle products and uploads it to Oracle's support systems. Consequently, you may access this information through your My Oracle Support account and Oracle can contact you if there are any security updates.

Note: The information collected by Security Updates is limited to configuration information. The data collected does not include personally identifiable information (with the exception of a local contact name in case of transmission problems). You may still use all licensed Oracle functionality if you decline to enable Security Updates.

You may choose not to be notified for any critical security updates. Simply leave all fields in the Configure Security Updates screen blank and click **Next** to continue.

The link to Oracle's policy page may not open if your default Web browser is an older version or not supported. Either update your browser or enter the following URL in your browser:

<http://www.oracle.com/support/policies.html>

E-mail

Specify your My Oracle Support e-mail address. If you do not have a My Oracle Support account, you may specify a personal e-mail address.

I wish to receive security updates via My Oracle Support

Select this check box if you want to receive security updates through My Oracle Support. Ensure that the e-mail address you entered in **Email** and the password you entered in **My Oracle Support Password** corresponds to your My Oracle Support user name and password, respectively.

If you specified a personal e-mail address in **Email**, then ensure that this check box is not selected.

My Oracle Support Password

Specify your My Oracle Support password.

Specify proxy server information

If you click **Next** and the Install Wizard cannot establish a direct connection to an outside network to send your My Oracle Support information, the **Specify proxy server information** window appears. Enter the following information:

- **Proxy Server:** The host name of your proxy server
- **Proxy Port:** The port number of your proxy server

- **Proxy Username:** The user name required to authenticate your proxy server, if required
- **Proxy Password:** The password required to authenticate your proxy server, if required
- **I want to remain uninformed of critical issues in my configuration:** Selecting this check box disables Security Updates. If you select this check box, you do not need to specify any proxy details.

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.

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 2.

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`

Note: The global database name may not contain a hyphen (-), and it may not start with a digit. It may contain only alphanumeric characters and the underscore (_).

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.

Template Information

Specify the template and memory to allocate for this Oracle Beehive installation:

Select Template

Select from one of the following templates:

- **Basic Server Template:** This is the default template. It will install and configure Oracle Beehive Release 2 against an existing database. The computer on which you are installing Oracle Beehive Release 2 must have at least 2 GB of RAM.
- **Basic Server and Client Template:** This template will install and configure Oracle Beehive Release 2, including Oracle Beehive Integration for Zimbra and Oracle Beehive Team Collaboration on the same computer. To use this template, your computer must have at least 3 GB of RAM.
- **Client Only Template:** This template will only configure Oracle Beehive Integration for Zimbra and Oracle Beehive Team Collaboration on your computer. An Oracle Beehive schema must already be installed in an existing database, you must have at least one Oracle Beehive application tier already installed, and your computer must have at least 2 GB of RAM.

Note: The Basic Server and Client and Client Only templates replace the option to install Oracle Beehive Integration for Zimbra in previous releases of Oracle Beehive.

Memory to Allocate

Specify the amount of RAM in GB to allocate for this Oracle Beehive installation. The amount you specify must be a whole number. The minimum amount depends on the template that you selected. In particular, the minimum amount for the Basic Server and Client templates is 2 GB and the minimum for the Basic Service and Client template is 3 GB. The maximum amount depends on the amount of RAM in your computer. For example, if you chose the Basic Server template and have 6 GB of RAM, you may specify 2, 3, 4, 5, or 6 GB of memory to allocate.

Enterprise Definition Information

Specify the following enterprise site information that identifies this installation of Oracle Beehive Release 2:

- [Enterprise](#)
- [Organization](#)
- [Site](#)
- [Site Key](#)
- [Confirm Site Key](#)
- [Instance](#)

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 2, 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 `..`).

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 2, an Oracle Beehive deployment may have only one site.

Site Key

If you are installing your first Oracle Beehive application tier, specify the site key for this site. If you install any additional Oracle Beehive application tiers, you will be prompted for this site key. The site key must have the following characteristics:

- Be between 8 and 30 characters long.
- Only contain characters from the database character set (AL32UTF8).
- Only contain any valid Unicode letter character, digit, period, hyphen, and underscore.
- Contain at least one upper case character, one lower case character, and one numeric character.
- May not start or end with a space.

Confirm Site Key

Confirm the site key that you specified in the previous field.

Instance

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`.

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 2 instances must have a unique name.

Existing Enterprise Information

Note: This screen will only appear if you are installing Oracle Beehive Release 2 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.

Specify the following information:

- **Enterprise:** Select an enterprise to which this Oracle Beehive installation will belong.
- **Site:** The site to which the selected enterprise belongs will be displayed.
- **Site Key:** Specify the site key of the indicated site.
- **Instance:** Specify a name for this Oracle Beehive installation that will distinguish it from others in the selected enterprise. This name has have the following characteristics:
 - Not empty
 - Contain no spaces
 - Contain only valid ASCII characters
 - Not longer than 63 characters
 - Unique among other Oracle products configured in the same computer

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 2.

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 2 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 users will have the same password.

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

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.

Note: If you are in preview mode, the **Install** button is disabled. If you want to install Oracle Beehive, repeatedly click **Back** until you reach the **Welcome** screen and deselect the **Preview Mode** option.

Alternatively, click the **Save** button to save your non-validated input to a response file.

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.

Note: 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 2 or to install and configure Oracle Beehive Release 2. 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.
 - **Template Information:** The template and memory to allocate for this installation. To change the details, go back to the [Template 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.

- **beeadmin Password:** The masked password for the beeadmin user, which is the administrative user for Oracle Beehive Release 2 instances. To change the details, go back to the [beeadmin 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:

- Pre-configuration Actions: Removes directories not needed
- Oracle Beehive Initialization: Runs the Oracle Application Server clone method
- Oracle Beehive Database Configuration - SOA schemas: Runs the Oracle Application Server Integration Repository Creation Assistant (IRCA) script to seed SOA schemas into the target database
- Base Platform Support Configuration: Runs all required Oracle Application Server configuration tools
- Disabling OC4J Home Instance: Reconfigures `opmn.xml` to not start up the OC4J home instance
- Oracle Beehive Database Configuration - Beehive schemas: Runs `schema_install.pl` to seed Oracle Beehive database schemas
- Oracle Beehive Framework Core Configuration: Performs core configuration with `BeehiveCoreConfiguration.pl`

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 the Oracle Beehive Installation Guide of your operating system.

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` or `oraInstall`.

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 2 Provisioning Application:

- [Specify Home Location](#)
- [Repository Database Administrator Password](#)
- [Software Library Configuration](#)
- [Beehive Software Image Archive](#)
- [Beekeeper Software Image Archive](#)
- [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 Grid Control database that is used as the repository for the Oracle Beehive Release 2 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.

Software Library Configuration

In the **Location** field, specify the full path of a directory that will contain software images (in particular, zipped, installable versions of Oracle Beehive products). This directory must reside in the same host as your Oracle Management Service.

You will only see this screen if the Install Wizard cannot find a software library location already configured for Oracle Management Service.

Beehive Software Image Archive

The Install Wizard automatically uploads a zipped, installable version of Oracle Beehive for the same operating system as the Install Wizard to the Oracle Enterprise Manager Grid Control central software library.

In this screen, you may optionally specify the locations of zipped, installable versions of Oracle Beehive for other operating systems, which the Install Wizard uploads to the central software library.

The location of this software image archive is normally *<Oracle Beehive installation media>/install/beehive_home.zip*.

Alternatively, you may upload Oracle Beehive software image archives from Oracle Enterprise Manager Grid Control.

Beekeeper Software Image Archive

The Install Wizard automatically uploads a zipped, installable version of Oracle Beekeeper for the same operating system as the Install Wizard to the Oracle Enterprise Manager Grid Control central software library.

In this screen, you may optionally specify the locations of zipped, installable versions of Oracle Beekeeper for other operating systems, which the Install Wizard uploads to the central software library.

The location of this software image archive is normally *<Oracle Beekeeper installation media>/install/beekeeper_home.zip*.

Alternatively, you may upload Oracle Beekeeper software image archives from Oracle Enterprise Manager Grid Control.

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 2 instance uses as a repository.
- **Software Library Configuration:** The full path of the directory that will contain software image archives.
- **Beehive Software Image Archive:** The locations of zipped, installable versions of Oracle Beehive for other operating systems.
- **Beekeeper Software Image Archive:** The locations of zipped, installable versions of Oracle Beekeeper for other operating systems.

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

Installation Progress

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
- Configure Software Library
- Package Beehive Software Image Archive
- Package Beekeeper Software Image Archive
- 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 "[Provisioning Oracle Beehive](#)" 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 2 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 2 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 2 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 2 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 2 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 the Oracle Beehive Installation Guide of your operating system.

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 2 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 Beekeeper Installation Help

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

Refer to ["Starting the Oracle Beekeeper Uninstall Wizard"](#) to uninstall Oracle Beekeeper.

Product Selection Sequence of Screens

The following screens appear before product selection:

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

Once you select a product type, you will proceed to ["Oracle Beekeeper Sequence of Screens"](#).

Welcome

The Welcome screen is the first screen of the Oracle Install Wizard. It will guide you through the installation and configuration of Oracle Beekeeper.

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 Beekeeper 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 Beekeeper 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.

Note: 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 Beekeeper.
- 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.

Select Product Type

Select which Oracle Beekeeper product you want to install:

- **Oracle Beekeeper:** This option installs Oracle Beekeeper on the computer from which you are running the Install Wizard.
- **Provisioning Application for Oracle Beekeeper:** This will allow you to install and configure Oracle Beekeeper 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 5 (10.2.0.5) or later.

Oracle Beekeeper Sequence of Screens

The following screens appear during the installation of Oracle Beekeeper:

- [Specify Home Location](#)
- [Prerequisite Checks](#)
- [Database Information](#)
- [Site Key Information](#)

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

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 Beekeeper 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 Beekeeper) meets all minimum requirements.

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

- Operating system certification
- Kernel parameters
- Recommended operating system packages
- Available swap space
- Recommended glibc version (Linux and UNIX-based operating systems)
- libdb.so.2 exists (UNIX-based operating systems)
- Sufficient file descriptors are allocated (Linux)
- Disk space
- In Microsoft Windows, ensuring that the Window user for Oracle Beekeeper installation has administrative privileges.

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

Note: 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.

Site Key Information

Enter the site key that you specified when you installed your first Oracle Beehive application tier.

Installation Summary

The Installation 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.

Note: 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 Beekeeper.
- **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, service name, and masked schema user password of the existing Oracle Beehive database that will be used by this installation process.
 - **Site Key:** The masked site key that you entered.

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 Beekeeper:

- Oracle Beekeeper Initialization: Runs the Oracle Universal Installer clone method
- Oracle Beekeeper Port Configuration: Ensures that the Oracle Beekeeper OC4J instance has dynamically allocated port values to ensure that no port conflicts occur with other processes after install
- Oracle Beehive Control OC4J Configuration: Ensures proper java arguments are configured in the standard `oc4j` command
- Oracle Beehive Control Service Creation: Establishes the Oracle Beekeeper service in the Oracle Beehive repository
- Starting OC4J Instance: Starts the OC4J container instance
- Oracle Beekeeper Framework Deployment: Deploys the Oracle Beehive Framework application into the home instance
- Oracle Beekeeper Application Deployment: Deploys the `beehivecontrol.ear` file into the home OC4J instance

- **Configuring OC4J Application Settings:** This modifies the OC4J related configuration files for Oracle Beekeeper operation
- **Stopping OC4J Instance:** Stops the OC4J instance
- **Starting OC4J Instance:** Starts the OC4J container instance

End of Installation

This screen indicates if the installation of Oracle Beekeeper was successful.

Starting the Oracle Beekeeper Uninstall Wizard

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

Oracle Beekeeper Uninstall Wizard Sequence of Screens

The following screens appear during the uninstallation of Oracle Beekeeper:

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

Select Oracle Home to Uninstall

This screen lists all the Oracle Beekeeper instances you have installed and their respective Oracle homes.

Select the Oracle Beekeeper 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 Beekeeper Uninstall Wizard must perform to uninstall your chosen Oracle Beekeeper 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 Beekeeper instance.

Note: On Microsoft Windows, if you receive an error message from the Oracle Beekeeper 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 Beekeeper instance is successful.

Provisioning Oracle Beehive

The Oracle Beehive Enterprise Deployment Procedure enables you to provision and deploy Oracle Beehive products from Oracle Enterprise Manager Grid Control. The Provisioning Advisor Framework (PAF) is used for provisioning of Oracle Beehive products.

You may deploy the following Oracle Beehive products with the Oracle Beehive Enterprise Deployment Procedure:

- Oracle Beehive
- Oracle Beehive for DMZ
- Oracle Beekeeper
- Oracle Coexistence Connector for Microsoft Exchange Server
- Oracle Coexistence Connector for Lotus Domino Server

This chapter covers the following topics:

- [Introduction to Oracle Beehive Enterprise Deployment Procedure](#)
- [Running Oracle Beehive Enterprise Deployment Procedure](#)
- [Uploading New Software Image Archive to Oracle Enterprise Manager Grid Control](#)

Introduction to Oracle Beehive Enterprise Deployment Procedure

The Oracle Beehive Enterprise Deployment Procedure consists of the following components:

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

Provisioning Advisor Framework

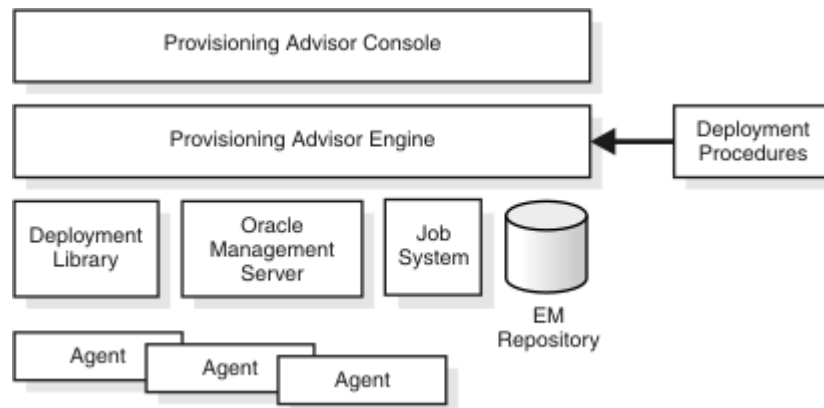
Provisioning Advisor Framework (PAF) provides a seamless process to integrate tasks that are associated with an application. The framework automates, orchestrates, and tracks the tasks of various applications, such as installers, upgrade assistants, and custom scripts.

PAF is built on top of the Enterprise Manager Grid Control structure. This structure comprises of Oracle Management Server (OMS), Job System, Oracle 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 6–1 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 computers. 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 Enterprise Deployment Procedure

Follow these steps to provision and deploy Oracle Beehive:

1. Select the **Deployments** tab.
2. Click **Deployment Procedures** (from the Deployment Procedure Manager section).
3. Select the radio button next to **Oracle Beehive Enterprise Deployment Procedure**.
4. Click the **Schedule Deployment** button.

Depending on which Oracle Beehive product you want to deploy, the Oracle Beehive Enterprise Deployment Procedure will present you with some of the following pages in which you enter required parameters:

1. [Deployment Targets](#)
2. [Upload](#)

3. [Configure Security Updates](#)
4. [Beehive Install Input](#)
5. [Beehive Templates](#)
6. [Beekeeper Install Input](#)
7. [Exchange/Domino Connector Install Input](#)
8. [Domino Install Input](#)
9. [Credentials/Schedule](#)
10. [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.

Deployment Targets

Select the targets on which you want to install an Oracle Beehive product. You may select more than one target, add a target to the list, and remove a target from the list.

Adding Targets

To add targets to the list of targets on this page, follow these steps:

1. Click the **Add** button.
2. In the **Select Target** window, search for available targets by specifying the following criteria. By default, the deployment procedure will search for all available targets:
 - **Type:** This is always set as **Host**.
 - **Host Name:** Specify the name of the host. Leave blank to search for hosts of any name. You may use the percent sign (%) and the asterisk (*) as wildcard replacements, for example %value, %value%, or *value%.
 - **Platform:** Specify **All** to search for targets of any supported platform. Specify a particular platform to search for targets of that platform.

Click **Go** to perform the search.

3. Select the targets you want to add and click the **Select** button to return to the **Deployment Targets** window.

Selecting Targets

Specify the following for each target that you want deploy an Oracle Beehive product:

- **Deployment Type:** Depending on the type of target, you may deploy one or more of the following:
 - **Oracle Beehive**
 - **Oracle Beehive for DMZ:** After deployment, you must configure DMZ instances from a non-DMZ Oracle Beehive instance
 - **Oracle Beekeeper**

- **Oracle Collaboration Coexistence Gateway:** At least one Oracle Coexistence Connector is required per Microsoft Exchange domain. To deploy this product, select the Oracle Coexistence Connector for Microsoft Exchange Server.
- **Oracle Coexistence Connector for IBM Lotus Domino Server:** Lotus Domino Server must reside in the specified deployment target
- **Home Location:** Specify the absolute path name of the directory in which you want to deploy the Oracle Beehive product.
- **Inventory Pointer:** This option is available only for UNIX-based operating systems.

By default, this field contains the absolute path name of the target host's central Oracle inventory pointer file. You may instead specify the absolute path name of a different inventory pointer file, which the deployment procedure will use as a private inventory for the Oracle Beehive product you are deploying.

Upload

If you have not already uploaded a zipped installable version of the Oracle Beehive product you are deploying to Oracle Beehive Enterprise Manager Grid Control Software Library, this page will prompt you to do so.

Linux and UNIX-Based Operating Systems

Upload the zipped installable version of the Oracle Beehive from *<Oracle Beehive installation media directory>/install/beehive_home.zip*.

Upload the zipped installable version of Oracle Beekeeper from *<Oracle Beekeeper installation media directory>/install/beekeeper_home.zip*.

Microsoft Windows

Upload the zipped installable version of Oracle Beehive from *<Oracle Beehive installation media directory>\install\beehive_home.zip*.

Upload the zipped installable version of Oracle Beekeeper from *<Oracle Beekeeper installation media directory>\install\beekeeper_home.zip*.

Configure Security Updates

Specify your My Oracle Support account details so that Oracle can notify you of any critical security updates.

Periodically, Security Updates automatically gathers configuration information of your installed Oracle products and uploads it to Oracle's support systems. Consequently, you may access this information through your My Oracle Support account and Oracle can contact you if there are any security updates.

Note: The information collected by Security Updates is limited to configuration information. The data collected does not include personally identifiable information (with the exception of a local contact name in case of transmission problems). You may still use all licensed Oracle functionality if you decline to enable Security Updates.

You may choose not to be notified for any critical security updates. Simply leave all fields in the Configure Security Updates page blank and click **Next** to continue. The deployment procedure will ask you, "Do you wish to remain uninformed of critical security issues in your configuration?" Click **No** to continue.

- **My Oracle Support Details:**

- **Email:** Specify your My Oracle Support e-mail address. If you do not have a My Oracle Support account, you may specify a personal e-mail address.
- **I wish to receive security updates via My Oracle Support:** Select this check box if you want to receive security updates through My Oracle Support. Ensure that the e-mail address you entered in **Email** and the password you entered in **My Oracle Support Password** corresponds to your My Oracle Support user name and password, respectively.

If you specified a personal e-mail address in **Email**, then ensure that this check box is not selected.

- **My Oracle Support Password:** Specify your My Oracle Support password.

- **Connection Details:** If you cannot establish a direct connection to an outside network to send your My Oracle Support information without a proxy server, then enter the following information:

- **Proxy Server:** The host name of your proxy server
- **Proxy Port:** The port number of your proxy server
- **Proxy Username:** The user name required to authenticate your proxy server, if required
- **Proxy Password:** The password required to authenticate your proxy server, if required

Beehive Install Input

The following describes the parameters for the **Beehive Install Input** page.

Note: If the operating system of your target host is Microsoft Windows, the Oracle Beehive Enterprise Deployment Procedure does not support non-ASCII data.

- **Database Information:** Specify the following details of an existing Oracle Database instance that will be used by this installation. Refer to "[Oracle Beehive Database Requirements](#)" in this guide for database requirements for Oracle Beehive Release 2.

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; the Oracle Beehive Enterprise Deployment Procedure will not verify it by trying to connect to each database host. Instead, the Oracle Beehive Enterprise Deployment Procedure constructs the connect descriptor with the database information you provide, then verifies if the connect descriptor can reach the database.

If you are installing Oracle Beehive in a high availability environment (you are installing multiple instances of Oracle Beehive) with the deployment procedure and you are using an Oracle RAC database, you may only use test certificates; you may not use self-signed 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 2, 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.

- **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 2, an Oracle Beehive deployment may have only one site.

- **Site Key:** If you are installing your first Oracle Beehive application tier, specify the site key for this site. If you install any additional Oracle Beehive application tiers, you will be prompted for this site key.
- **Confirm Site Key:** Confirm the site key that you specified in the previous field.
- **Instance:** 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 2 instances must have a unique name.

- Schema Credentials
 - **Password:** 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.
- beeadmin Credentials
 - **Password:** Specify the password for the beeadmin user. The `oc4jadmin` users will have the same password.
 - **Confirm Password:** Re-enter the password you entered in the previous field.

Beehive Templates

The following describes the parameters for the **Beehive Templates** page:

- **Select Template:** For each deployment target, select from one of the following templates:

- **Basic Server Template:** This is the default template. It will install and configure Oracle Beehive against an existing database. The deployment target must have at least 2 GB of RAM.
- **Basic Server and Client Template:** This template will install and configure Oracle Beehive, including Oracle Beehive Integration for Zimbra and Oracle Beehive Team Collaboration on the same computer. The deployment target must have at least 3 GB of RAM.
- **Client Only Template:** This template will only configure Oracle Beehive Integration for Zimbra and Oracle Beehive Team Collaboration on the deployment target. The deployment target must have an Oracle Beehive schema installed in an existing database, at least one Oracle Beehive application tier, and at least 2 GB of RAM.
- **Allocate Mem (GB):** Specify the amount of RAM in GB to allocate for this deployment. The amount you specify must be a whole number. The minimum amount depends on the template that you selected. In particular, the minimum amount for the Basic Server and Client templates is 2 GB and the minimum for the Basic Service and Client template is 3 GB. The maximum amount depends on the amount of RAM in your computer. For example, if you chose the Basic Server template and have 6 GB of RAM, you may specify 2, 3, 4, 5, or 6 GB of memory to allocate.

Beekeeper Install Input

The following describes the parameters for the **Beekeeper Install Input** page:

- **Beehive Database Information:** Specify the following details of an existing Oracle Database instance that will be used by this deployment. Refer to "[Oracle Beehive Database Requirements](#)" in this guide for database requirements for Oracle Beehive.

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; the deployment procedure will not verify it by trying to connect to each database host. Instead, the deployment procedure constructs the connect descriptor with the database information you provide, then verifies if the connect descriptor can reach the database.

- **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 Beekeeper"](#) for more information. Refer to ["Database Information"](#) in ["Oracle Beekeeper Installation Help"](#) for more information about the service name.

- **Password:** Enter the password for the Oracle Beehive database schema that is stored in the database whose information you have specified in this screen.
- **Site Key:** Enter the site key that you specified when you installed your first Oracle Beehive application tier.

Exchange/Domino Connector Install Input

The following describes the parameters for the **Exchange/Domino Connector Install Input** page:

- **Oracle Beehive Server details:** Specify the following details of the Oracle Beehive application tier that is running Oracle Coexistence Connector.
 - **Host:** Enter the host name, real or virtual, used to access your Oracle Beehive installation.
 - **Port:** Enter the HTTP port used to access your Oracle Beehive installation.
 - **HTTPS:** Select this option if you want to use an HTTPS connection
 - **Partnership Key:** The partnership key is a password that will be used for authentication between Oracle Coexistence Connector and Oracle Beehive Coexistence Service.
Enter any string to be used as partnership site key.
 - **Confirm Partnership Key:** Re-enter the password you entered in the previous field.
- **Password for "oc4jadmin" administrative user:** Specify and confirm the password for the oc4jadmin administrative user. The oc4jadmin user will be exposed if you set up monitoring with Oracle Enterprise Manager Grid Control.
 - **Password:** Enter the password for the oc4jadmin user.

Note: Password Restrictions: The password of the oc4jadmin 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 (_).

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

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,
-

- **Confirm Password:** Re-enter the password you entered in the previous field.

Domino Install Input

The following describes the parameters for the **Domino Install Input** page:

- **Domino Server and User Info:** Coexistence Connector for Domino requires a Domino user account with administrative privileges to access the Lotus Domino server. This is a Domino user and password account (not a Windows user account).
 - **Domino Host HTTP Port:** Specify the HTTP port number of your Domino server.
 - **Domino Admin User Name:** Specify the user name of your Domino server administrator's account.
 - **Domino Admin User Password:** Specify the password of the user you specified in the previous field.
- **Domino Foreign Domain Name:** All coexisting Oracle Beehive users will reside or be created in this foreign domain as foreign users.
 - **Foreign Domain Name:** Specify the Domino foreign domain name. This name must be between one and 64 characters long. The only valid characters are a-z, A-Z, 0-9, and the underscore (_).

Credentials/Schedule

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.
 - **Specify Credentials for:** Specify the Oracle homes that the deployment procedure will access with the credentials (username and password) that you will specify in the following fields.
 - * **Username, Password, and Confirm Password:** Specify the user name 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.

Uploading New Software Image Archive to Oracle Enterprise Manager Grid Control

You may upload a new software image archive (an installable version of an Oracle Beehive product) to the central software library of Oracle Enterprise Manager Grid Control. Oracle Beehive Provisioning Application uses this software image archive to install Oracle Beehive products.

To upload a new software image archive to the central library of Oracle Enterprise Manager 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 Enterprise Manager Grid Control.
2. In the TreeTable, expand the **Components** entry until you see **Oracle Beehive Software Image Archive** or **Oracle Beekeeper Software Image Archive**. Select the software image archive you want to replace with a new one. 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 software image archive:
 - **Oracle Beehive**
 - **Linux and UNIX-Based operating systems:** *<Oracle Beehive installation media>/install/beehive_home.zip*
 - **Microsoft Windows:** *<Oracle Beehive installation media>\install\beehive_home.zip*
 - **Oracle Beekeeper**
 - **Linux and UNIX-Based operating systems:** *<Oracle Beekeeper installation media>/install/beekeeper_home.zip*
 - **Microsoft Windows:** *<Oracle Beekeeper installation media>\install\beekeeper_home.zip*
4. Click **Finish**.

Installing Oracle Beehive in Silent Mode (Non-Interactive)

This chapter 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 chapter 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 and Config Wizards](#)
- [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	STATUS

11.1.0.7.0	OPEN

```
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/beehive_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: 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 and Config Wizards

This process consists of two phases:

- an installation-only (or software-only) phase that installs Oracle Beehive by running the Oracle Beehive Install Wizard.
- 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
BEEGMT_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
- Oracle Beehive Provisioning Application. (You may not upgrade any version of Oracle Beehive Provisioning Application earlier than 1.2.1.0.0.)
- Oracle Beehive for DMZ
- Oracle Beekeeper

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: `beehive_upgrade_response_template.xml`
- Oracle Beehive Provisioning Application: `provplugin_upgrade_response_template.xml`
- Oracle Beehive for DMZ: `dmz_upgrade_response_template.xml`
- Oracle Beekeeper: `beekeeper_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 chapter describes command-line options for the Oracle Beehive Install Wizard and Config Wizard.

Usage

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

The Install Wizard is located in the installation media. The full path of the Config Wizard is `<Oracle home>/beehive/oobwiz/configWizard`.

Options

The following table describes the command line options for both the Install Wizard and the Config Wizard:

Table 8–1 Install and Config Wizard Command-Line 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 <code>oracle.bee_allProducts</code> .
-force	Allows silent mode installation into a non-empty directory. Valid for install time use only.
-forceOHInventoryUninstall <i>Oracle_home</i>	Removes the specified Oracle home from the Oracle inventory.
-generateResponseTemplate <i>directory</i>	Generates response files (for Oracle Beehive installation, Oracle Beehive Provisioning Application, and Oracle Beehive uninstallation) in the specified directory or the user's home directory if no directory is specified.
-help	Outputs help about the wizard's command-line arguments.

Table 8–1 (Cont.) Install and Config Wizard Command-Line 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 Oracle Solaris SPARC on (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>	<p>Filters messages that have a lesser priority level than <i>level</i>.</p> <p>Valid levels are severe, warning, info, config, fine, finer, and finest.</p>
<code>-noConfig</code>	Installs the software only; does not perform any configuration; valid for install time use only.
<code>-noconsole</code>	Windows only: Suppresses display of messages to the console.
<code>-noSplash</code>	Suppresses the wizard splash screen.
<code>-nowait</code>	<p>Windows only: Does not prompt the user for exit confirmation.</p> <p>This option is useful if you call the wizard from a batch file and do not want the yes/no confirmation prompt at the end of execution.</p>
<code>-patch</code>	Runs the wizard in patch set mode.
<code>-prereq_config_loc <i>location</i></code>	Specifies the fully qualified path to the prerequisite directory that contains the <code>prerequisite.properties</code> file.
<code>-prereqcheck</code>	<p>Runs the prerequisite checker only.</p> <p>You may run the Install or Config Wizard with this option to determine if your computer meets the Oracle Beehive's system requirements before installation or configuration.</p> <p>In addition, to determine if your computer has enough disk space. Specify the directory in which you want to install or configure the Oracle Beehive product with the <code>-target_loc</code> option.</p>
<code>-record</code>	<p>Records the user's response from the installation wizard to a response file.</p> <p>Specify the name of the file to be created with the <code>-responseFile</code> option.</p>
<code>-responseFile <i>path</i></code>	Specifies the response file and path to use.
<code>-silent</code>	Installs Oracle Beehive in silent mode operation. In addition, you must 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> .

Table 8–1 (Cont.) Install and Config Wizard Command-Line Options

Option	Description
-uninstall	<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 -silent and -responseFile <i>uninstall_responsefile</i>.</p> <p>Windows only: This option will work only from <i>setup.exe</i> on the installation media.</p>
-version	<p>Outputs the version of the framework of the wizard.</p>
-waitforcompletion	<p>Causes the installer process to execute in the foreground; it will not return until execution completes.</p> <p>This option is useful if you call the wizard from a shell script or batch file and want to use the value that the wizard returns when it exits.</p>

Oracle Beehive Uninstall Wizard

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

- Oracle Beehive Release 2 (2.0)
- Oracle Beehive Provisioning Application
- Oracle Beehive for DMZ

This chapter covers the following topics about uninstalling Oracle Beehive:

- [Starting the Oracle Beehive Install Wizard](#)
- [Oracle Beehive Uninstall Wizard Sequence of Screens](#)
- [Uninstalling Provisioning Application for Oracle Beehive or Oracle Beekeeper](#)
- [Uninstalling Multiple Instances](#)
- [Manually Deleting Oracle Beehive Tablespaces and Datafiles](#)
- [Uninstalling Corrupted Installation](#)

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 2 (2.0) 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 2 (2.0) and it is the last instance configured in your database.

Select 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 Provisioning Application for Oracle Beehive or Oracle Beekeeper

Uninstall Oracle Beehive Provisioning Application or Oracle Beekeeper Provisioning Application with the Oracle Beehive Universal Installer:

1. Run the Oracle Beehive Universal Installer from the Oracle Management Service home:
`<Oracle Management Service home>/oui/bin/runInstaller`
2. Click **Deinstall Products**.
3. In the **Oracle Homes** tree, expand the `<OMS_HOME>` node to view all installed components. Select the provisioning application you want to remove and click the **Remove** button.

Note: You cannot uninstall the provisioning application with the Oracle Beehive or Oracle Beekeeper Uninstall Wizard.

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

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 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
 - 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_INDEX 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.

Manually Deleting Existing BEE_CODE Schema

To manually delete old BEE_CODE schemas, run the following SQL*Plus commands as a user with SYSDBA privileges on any one of the mid-tiers:

bee_code_xxx schemas:

```
cd $ORACLE_HOME/bee hive/db
```

```
$ORACLE_HOME/perl/bin/perl schema_deinstall.pl --oracle_home $ORACLE_HOME
--schema_name <schema_to_be_dropped> --new_status deinstalled
```

Uninstalling Corrupted Installation

To get system object of the corrupted midtier, run the following beectl command from the remote midtier:

```
beectl list_components --type BeehiveInstance
```

Follow these steps to uninstall a corrupted Oracle Beehive installation:

1. Run the Install Wizard or the Config Wizard with the `-forceOHInventoryUninstall` option. Note that you may run this command from the installation media or the Oracle Beehive home directory. Run one of the following commands depending on your operating system:

```
<Oracle Beehive home>/beehive/oobwiz/configWizard -forceOHInventoryUninstall
<Oracle Beehive Home>
runInstaller -forceOHInventoryUninstall
setup.exe -forceOHInventoryUninstall
```

2. Manually remove the Oracle Beehive home directory.
3. Clean up the Oracle Beehive configuration remotely by running the `beectl delete_remote_home_instance` command.

Note that you must run this command from a remote Oracle Beehive instance.

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
```

Exporting Configuration Data

You may export the entire system model configuration data into an XML file for any given configuration version with the `beectl export_configuration_data` command.

You may provide your Oracle support representative the output from this command. This will help your representative investigate any problems with your Oracle Beehive deployment faster. You may also use the output of this command to compare the configuration between different Oracle Beehive deployments, such as between a test and a production deployment.

Note that secure values do not appear in the output for security reasons.

HTTP Server mod_oc4j Continues Trying to Route to OC4J after Being Disconnected

If you suddenly shut down an application tier, you may receive the following error messages in your Oracle HTTP Server error log file (typically named `error_log`):

```
[Thu Sep  4 12:59:07 2008] [error] [client 140.87.120.38]
[ecid:1220558336:144.25.34.13:3454:0:2430,0] mod_oc4j: request to OC4J
myhost.example.com:12503 failed: Connect failed (errno=113)
```

...

```
[Thu Sep  4 13:02:40 2008] [error] [client 140.87.120.38]
[ecid:1220558560:144.25.34.13:3802:0:2395,0] mod_oc4j: request to OC4J
myhost.oracle.com:12503 failed: Connect failed (errno=111)
```

In this situation, the `mod_oc4j` module has not been notified that the application tier it is trying to connect to has been shut down. Consequently, it will repeatedly attempt to connect to the application tier until it fails a certain number of times, which is specified by the `mod_oc4j` parameter `MaxErrors`.

The default value for `MaxErrors` is 100. You may change the value of this parameter to a lower value (such as 1) by following these steps:

1. Edit the file `<Oracle home>/beehive/conf/scripts/httpd.conf.tmp` and add the line `Oc4jSet MaxErrors 1` in the `<IfModule mod_oc4j.c>` section:

```
<IfModule mod_oc4j.c>
    Oc4jSet MaxErrors 1
</IfModule>
```

2. Run the following command to regenerate the file `<Oracle home>/Apache/Apache/conf/mod_oc4j.conf` (which contains the `mod_oc4j` parameter):

```
beectl modify_local_configuration_files --restart_needed false
```

3. Run the following command to restart Oracle HTTP Server:

```
<Oracle home>/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

Refer to My Oracle Support Note 468325.1 "HTTP Server mod_oc4j Continues Trying To Route To OC4J After Node Is Disconnected" for more information.

When running the stress tool, BEEAPP OC4J crashes due to Out Of Memory Error

During post install or patch install on the first midtier, follow the below mentioned steps:

1. Get all BSOM Database objects, this command lists all database objects in the system.

```
beectl list_components --type Database
```

2. For each Database object listed in step 1, run the following command:

```
beectl append_value --component <Database-BSOM-OBJECT-ID>:AffinityPool --name
ConnectionProperties --value oracle.jdbc.FreeMemoryOnEnterImplicitCache:true
beectl append_value --component <Database-BSOM-OBJECT-ID>:DefaultNonXaPool
--name ConnectionProperties --value
```

```
oracle.jdbc.FreeMemoryOnEnterImplicitCache:true
```

<Database-BSOM-OBJECT-ID> is the Database component ID listed in step 1.

3. Activate the configuration:

```
beectl activate_configuration  
beectl modify_local_configuration_files
```

4. Restart the midtier and then install the patch on the second midtier.

When cloning application tier in a multi midtier environment, iAS cloning fails

When cloning the application tier in a multi midtier environment, the Application Server cloning fails.

The workaround for this issue is to edit the \$OH/oui/oraparam.ini file and replace the following values:

- %JRE_LOCATION% - Change the path to point to JRE directory location, for example: ../../jdk/jre
- %JRE_MEMORY_OPTIONS% - Change the value to "-mx160m"

After changing the values of JRE_LOCATION and JRE_MEMORY_OPTIONS, execute beectl clone_mdtier command on the new machine.

For more information, see [Application Tier Cloning](#).

Part II

Oracle Beehive Upgrade Procedures

The Part II of the Installation Guide describes how to upgrade your Oracle Beehive deployment to Oracle Beehive Release 2 (2.0). To install Oracle Beehive, refer to [Part I, "Oracle Beehive Installation"](#). To configure Oracle Beehive, refer to [Part III, "Oracle Beehive Post-Installation Configuration"](#). To install Oracle Beehive in a high availability environment, refer to [Part IV, "Oracle Beehive High Availability Configuration"](#).

This part contains the following chapters:

- [Upgrading Oracle Beehive Overview](#)
- [Upgrading Oracle Beehive](#)
- [Upgrading Oracle Beekeeper](#)
- [Oracle Beehive By Clone Patching](#)

Upgrading Oracle Beehive Overview

This chapter describes steps to perform before upgrading your Oracle Beehive deployment to Oracle Beehive Release 2 (2.0), the order in which you should upgrade Oracle Beehive products, and other procedures to perform after upgrading.

This section covers the following topics:

- [Upgrade Options](#)
- [Before Upgrading](#)
- [Upgrade Sequence](#)
- [Upgrading Multiple Oracle Beehive Application Tiers](#)
- [Post-Upgrade Procedures](#)
- [Upgrading Oracle Beehive by Clone Patching](#)
- [Creating Batches For Use With add_search_recovery_scope Command](#)

Upgrade Options

Oracle Beehive 2.0.1.8.0 is available as upgrade, using the Oracle Universal Installer; or, as the 2.0.1.8.0 Cumulative Patch Set, which applies updates to pre-existing version installation as shown in the [Table 11–1, "Oracle Beehive Upgrade Options for 2.0.1.8.0"](#).

For information about installing the 2.0.1.8.0 Cumulative Patch Set, refer to the *Oracle Beehive 2.0.1.8.0 Cumulative Patch Set Readme* at the following location:

<http://www.oracle.com/pls/bee2/homepage>

For the full install/upgrade, you can use the *Oracle Beehive Installation Guide Release 2.0* for your platform, as well as the installer integrated help.

The following table shows the possible upgrade and patch options that are available:

Table 11–1 Oracle Beehive Upgrade Options for 2.0.1.8.0

Your current Oracle Beehive deployed version	Upgrade path option(s)
1.5.1.0.0, 1.5.1.2.0, 1.5.1.3.0, 1.5.1.4.0	<ul style="list-style-type: none"> ■ Patch to 1.5.1.5.0, then use 2.0.1.2.0 OUI installer ■ then apply 2.0.1.8.0 Cumulative Patch Set
1.5.1.5.0, 1.5.1.5.1, or 1.5.1.5.2	<ul style="list-style-type: none"> ■ Use 2.0.1.2.0 OUI installer ■ then apply 2.0.1.8.0 Cumulative Patch Set

Table 11–1 (Cont.) Oracle Beehive Upgrade Options for 2.0.1.8.0

Your current Oracle Beehive deployed version	Upgrade path option(s)
2.0.1.0.0, 2.0.1.1.0, 2.0.1.2.0, 2.0.1.2.1, 2.0.1.3.0, 2.0.1.4.0, 2.0.1.5.0, 2.0.1.6, and 2.0.1.7	Apply 2.0.1.8.0 Cumulative Patch Set.

Note: You can determine your Oracle Beehive version number by running the beectl version command from any Oracle Beehive Oracle home.

Before Upgrading

Perform or ensure the following before upgrading to Oracle Beehive Release 2 (2.0):

- [Upgrade Database to Supported Version](#)
- [Ensure Passwords for Active Code and Data Schemas Are Same](#)
- [Adjust XmppTimerKeepAliveTime Configuration Parameter](#)
- [Analyze Application Tiers](#)
- [Rollback Oracle Application Server Critical Patch Update](#)
- [Ensure tnsnames.ora File Exists](#)
- [Export Configuration Data](#)
- [Prepare Oracle Beehive Integration for Zimbra for Upgrade](#)
- [Configure Zero Downtime Upgrade](#)
- [Shut down All Oracle Beehive Instances](#)

Upgrade Database to Supported Version

If your Oracle Beehive deployment uses an Oracle Database version earlier than 11.1.0.7, then upgrade it to a version that Oracle Beehive Release 2 (2.0) supports. Refer to "[Oracle Beehive Database Requirements](#)" for more information.

Ensure Passwords for Active Code and Data Schemas Are Same

Ensure that the passwords for the active Code and Data schemas are the same. If not, change the password before proceeding with the upgrade.

Adjust XmppTimerKeepAliveTime Configuration Parameter

Ensure the value of the configuration parameter XmppTimerKeepAliveTime is 10 or less. Run the following command to obtain the value of XmppTimerKeepAliveTime:

```
beectl list_properties --component _xmppservlet --name XmppTimerKeepAliveTime
```

```
-----+-----
Property name      | Property value
-----+-----
XmppTimerKeepAliveTime | 5
-----+-----
```

If the value of `XmppTimerKeepAliveTime` is greater than 10, set it to 5 (the default value) with the `beectl modify_property` command. After that, run the command `beectl activate_configuration` to commit changes to the configuration.

Analyze Application Tiers

Note: You do not have to perform this step in the following situations:

- You have installed (but not upgraded to) Oracle Beehive Release 1 (1.4).
 - You have installed Oracle Beehive Release 1 (1.2) or Release 1 (1.3) and have never cloned any application tiers or sites.
 - You have cloned an application tier or site only after upgrading to Oracle Beehive Release 1 (1.4).
-
-

Before upgrading Oracle Beehive Release 1 (1.5.x) to Oracle Beehive Release 2 (2.0), analyze each of your Oracle Beehive application tiers by running the command `beectl clone_preparation` on each of them. For more information about running this command, refer to ["Step 4: Call beectl clone_preparation Command"](#) in ["Cloning Oracle Beehive Application Tiers and Sites"](#). This command creates a text file that contains the names of files in the source Oracle home to be copied for cloning to the target location. You will not need this file to upgrade Oracle Beehive. However, if the `beectl clone_preparation` command fails for a particular application tier, you will not be able to upgrade it. You must uninstall any application tier where the `beectl clone_preparation` command fails before upgrading your Oracle Beehive deployment.

Rollback Oracle Application Server Critical Patch Update

If you applied an Oracle Application Server Critical Patch Update (CPU) patch to any of your Oracle Beehive Release 1 (1.5.x) application tiers, follow the steps described in My Oracle Support Note 735631.1, "Symbol Referencing Error on nzospRandNum When AS Patchset Applies Patch 4601861 (CPU Patch Previously Applied)." These steps involve rolling back previously applied CPU patches and applying the latest CPU patch for the new patchset version.

Ensure tnsnames.ora File Exists

If you are upgrading an Oracle Beehive Release 1 (1.5.x) application tier that has been upgraded from an Oracle Beehive Release 1 (1.3.1) application tier or earlier, ensure the file `<Oracle Beehive home>/network/admin/tnsnames.ora` exists before upgrading your Oracle Beehive Release 1 (1.5.x) application tier.

The `tnsnames.ora` file must contain an entry that specifies the TNS (Transport Network Substrate) identifier of BEEDB and the connection information of the database used by your Oracle Beehive deployment.

The following is an example of this entry (line breaks have been inserted for clarity):

```
BEEDB =
  (DESCRIPTION= (ADDRESS_LIST= (ADDRESS= (PROTOCOL=TCP) (HOST=mydb.example.com)
    (PORT=1521))) (CONNECT_DATA= (SERVICE_NAME=mysevice.example.com)))
```

Use one of the following methods to obtain the TNS identifier of BEEDB:

- Run the command `beectl list_bootstrap_configuration` and look for the `ConnectionString` property (the `--format` option is optional):

```
beectl list_bootstrap_configuration --format xml
```

```
<?xml version="1.1" encoding="UTF-8"?>
<beectl-output resultset="table">
  <row>
    <column name="Property Name">ConnectionString</column>
    <column name="Property Value">
      (DESCRIPTION=(ADDRESS_LIST=
        (ADDRESS=(PROTOCOL=TCP)
          (HOST=mydb.example.com) (PORT=1521)))
        (CONNECT_DATA=(SERVICE_NAME=beedb.example.com)))
    </column>
  </row>
```

- Run the command `beectl list_properties --component _CURRENT_SITE:Database` and look for the `ConnectDescriptor` property (the `--format` option is optional):

```
beectl list_properties --component _CURRENT_SITE:Database --format xml
```

```
...
<row>
  <column name="Property name">ConnectDescriptor</column>
  <column name="Property value">
    (DESCRIPTION=(ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=TCP)
        (HOST=mydb.example.com) (PORT=1521)))
      (CONNECT_DATA=(SERVICE_NAME=beedb.example.com)))
  </column>
</row>
```

Export Configuration Data

Before upgrading Oracle Beehive Release 1 (1.5.x) to Oracle Beehive Release 2 (2.0), export configuration data of your Oracle Beehive Release 1 (1.5.x) deployment that you want to preserve for future reference. In particular, you may want to preserve the configuration data of your Oracle Beehive deployment before you upgrade it.

Use the `beectl export_configuration_data` command to export configuration data.

Although the upgrade process does not remove or delete configuration data, it does not upgrade it to the new configuration data structure that Oracle Beehive Release 2 (2.0) uses. Consequently, you cannot access configuration data of previous versions of Oracle Beehive with the `beectl` command; contact your Oracle representative if you require access to this data.

Prepare Oracle Beehive Integration for Zimbra for Upgrade

Depending on which version and where you installed Oracle Beehive Integration for Zimbra, follow the steps in these sections:

- [Upgrading Oracle Beehive Release 1 \(1.5.x\) with Oracle Beehive Integration for Zimbra Version 1.5.x](#)
- [Upgrading Previously Upgraded Oracle Beehive Integration for Zimbra Instance](#)

Upgrading Oracle Beehive Release 1 (1.5.x) with Oracle Beehive Integration for Zimbra Version 1.5.x

If you are upgrading Oracle Beehive Release 1 (1.5.x) with Oracle Beehive Integration for Zimbra version 1.5.x registered in the same Oracle Inventory location, follow these steps before upgrading Oracle Beehive and Oracle Beekeeper to version 2.0. (Note that these steps still apply if you previously upgraded your Oracle Beehive or Oracle Beehive Integration for Zimbra instances to version 1.5.x):

1. Backup the file *<Oracle Beehive Integration for Zimbra home>/beehive/oobwiz/configWizard.properties*.
2. Edit the file *<Oracle Beehive Integration for Zimbra home>/beehive/oobwiz/configWizard.properties* as follows:

Find the line that begins with `InstallType=` and perform one of the following actions depending on your Oracle Beehive deployment:
 - If you chose the Basic Server or Basic Server and Client template when installing Oracle Beehive (only Oracle Beehive or Oracle Beehive and Oracle Beehive Integration for Zimbra reside in the computer), ensure the line is `InstallType=Beehive`.
 - If you chose the Client Only template when installing Oracle Beehive, ensure the line is `InstallType=Client`.
3. Perform any other required tasks before upgrading Oracle Beehive as described in this module, then upgrade Oracle Beehive to version 2.0.
4. After upgrading Oracle Beehive to version 2.0 but before upgrading Oracle Beehive Integration for Zimbra to version 2.0, copy the version of *configWizard.properties* you backed up to *<Oracle Beehive Integration for Zimbra home>/beehive/oobwiz*.

Alternatively, edit the file *<Oracle Beehive Integration for Zimbra home>/beehive/oobwiz/configWizard.properties* and change the line `InstallType=` to `InstallType=Client`.
5. Upgrade Oracle Beehive Integration for Zimbra to version 2.0.

Upgrading Previously Upgraded Oracle Beehive Integration for Zimbra Instance

If you are upgrading Oracle Beehive Integration for Zimbra version 1.5.x to version 2.0, perform the following steps before upgrading:

1. Backup the contents of the directory *<Oracle Beehive Integration for Zimbra home>/inventory/ContentsXML/ConfigXML*.
2. Retrieve a list of all the XML files that begin with `Beehive` from the directory *<Oracle Beehive Integration for Zimbra home>/inventory/ContentsXML/ConfigXML*:

```
ls -l Beehive*
```

```
BeehiveAggregate.<version number>.xml
BeehiveConfig.<version number>.xml
BeehiveDeconfig.<version number>.xml
```

From this list, delete all files **except** those whose `<version number>` is `1_5_1_0_0`.

For example, the folder `ConfigXML` may contain the following files on your system:

```
ls -l Beehive*
```

```
BeehiveAggregate.1_4_1_0_0.xml
BeehiveAggregate.1_4_3_0_0.xml
BeehiveAggregate.1_5_1_0_0.xml
BeehiveConfig.1_4_1_0_0.xml
BeehiveConfig.1_4_3_0_0.xml
BeehiveConfig.1_5_1_0_0.xml
BeehiveDeconfig.1_4_1_0_0.xml
BeehiveDeconfig.1_4_3_0_0.xml
BeehiveDeconfig.1_5_1_0_0.xml
```

After deleting the files that **do not** contain the version number 1_5_1_0_0, the folder ConfigXML should contain the following files:

```
ls -l Beehive*
```

```
BeehiveAggregate.1_5_1_0_0.xml
BeehiveConfig.1_5_1_0_0.xml
BeehiveDeconfig.1_5_1_0_0.xml
```

3. Edit the file *<Oracle Beehive Integration for Zimbra home>/beehive/install/beeStart.pl* and comment out the line that launches the `owsmInstallProperties` subroutine.

Search for the following lines in the file `beeStart.pl`:

```
# prepare owsm properties file
&owsmInstallProperties();
```

Add the number sign (#) to the beginning of the line `&owsmInstallProperties()`;

```
# prepare owsm properties file
# &owsmInstallProperties();
```

Configure Zero Downtime Upgrade

Note: This section only applies to patches for Oracle Beehive Release 2 (2.0).

Oracle Beehive Release 2 (2.0) comes with Zero Downtime Upgrade (ZDU), which minimizes the amount of downtime required for upgrading or applying patches to Oracle Beehive.

After upgrading to Oracle Beehive Release 2 (2.0) Patch and before upgrading or applying a newer patch to Oracle Beehive, perform the following steps:

- [Configure Oracle Data Pump](#)
- [Disable User Directory Services Synchronization](#)

Configure Oracle Data Pump

Part of the upgrade or patching process involves updating code objects, which are stored in its own code schema in the Oracle Beehive database. The ZDU process first clones the code schema, then updates the cloned schema. This allows users in a multi-application tier environment to continue using Oracle Beehive during the upgrade or patching process; updated application tiers would use the cloned, updated code schema while those tiers that have not been updated would use the original code schema.

The code schema cloning process uses Oracle Data Pump technology, which enables very high-speed movement of data and metadata from one database to another. You must configure Oracle Data Pump before upgrading or patching Oracle Beehive.

During the upgrade or patching process, Oracle Beehive uses two database directory objects named BEEHIVE_DATA_PUMP and BEEHIVE_DATA_PUMP_LOG, which Oracle Beehive uses as the Oracle Data Pump data directory and log file directory, respectively. Depending on whether your Oracle Beehive deployment uses Oracle RAC nodes or not, either perform ["Defining Oracle Data Pump Directories for Non-Oracle RAC Deployments"](#) or ["Defining Oracle Data Pump Directories for Oracle RAC Deployments"](#). Afterwards, perform ["Backing up and Deleting Oracle Data Pump Log Files"](#).

Defining Oracle Data Pump Directories for Non-Oracle RAC Deployments If your Oracle Beehive deployment does not use Oracle RAC, follow these steps to define the Oracle Data Pump directories BEEHIVE_DATA_PUMP and BEEHIVE_DATA_PUMP_LOG:

1. Create two directories on the computer that is hosting your database, one that will store Oracle Data Pump data and another Oracle Data Pump log files. Use regular file system commands, like `mkdir`, to create these directories.

ASM Note: If your database is using Oracle Automatic Storage Management (ASM), you must create the data directory in a new or existing disk group. Use the ASMCMD command-line utility as follows:

```
ASMCMD> mkdir +DISKGROUP1/bee hive_data_pump_directory
```

DISKGROUP1 is either an existing or new disk group. *bee hive_data_pump_directory* is the name of the data directory you are creating.

You must not create the log file directory in ASM; it must be a regular directory in the file system of the computer that is hosting your database.

Refer to "ASM Command-Line Utility" in *Oracle Database Storage Administrator's Guide* for more information about ASMCMD.

2. Run the following SQL*Plus commands from the computer hosting your database or from any Oracle RAC node as a DBA user or a user with the CREATE ANY DIRECTORY privilege:

```
SQL> CREATE OR REPLACE DIRECTORY BEEHIVE_DATA_PUMP AS '<Oracle Data Pump data
directory>';
SQL> GRANT READ, WRITE ON DIRECTORY bee hive_data_pump TO bee_code;
SQL> GRANT READ, WRITE ON DIRECTORY bee hive_data_pump TO bee_data;
SQL> CREATE OR REPLACE DIRECTORY BEEHIVE_DATA_PUMP_LOG AS '<Oracle Data Pump
log directory>';
SQL> GRANT READ, WRITE ON DIRECTORY bee hive_data_pump_log TO bee_code;
SQL> GRANT READ, WRITE ON DIRECTORY bee hive_data_pump_log TO bee_data;
```

bee_code and *bee_data* are the names of the Oracle Beehive code schema and data schema, respectively. To retrieve the names of these schemas, run the command `beectl list_schemas`.

Refer to "CREATE DIRECTORY" in *Oracle Database SQL Language Reference* for more information.

Defining Oracle Data Pump Directories for Oracle RAC Deployments If your Oracle Beehive deployment uses Oracle RAC, then ensure that all your Oracle RAC nodes can access both Oracle Data Pump directories.

Note: If your environment uses the OCFS2 file system, and that file system is shared between all Oracle RAC nodes, then follow the steps described in ["Defining Oracle Data Pump Directories for Non-Oracle RAC Deployments"](#). Ensure that you create the Oracle Data Pump data and log directories in the same OCFS2 file system that your Oracle RAC nodes can access.

You do not need a special connection string as described in the following steps for Oracle RAC deployments since all Oracle RAC nodes can access the data and log directories in an OCFS2 file system.

Follow one of the followings steps to create and specify these directories:

- Follow the steps as described in ["Defining Oracle Data Pump Directories for Non-Oracle RAC Deployments"](#).
 - If you are using ASM, then ensure that all your Oracle RAC nodes can access BEEHIVE_DATA_PUMP_LOG.

Use regular file system commands, like `mkdir`, to create the same directory (you must use the same directory path) on the local disk of each Oracle RAC node.
 - If you are not using ASM, the Oracle Beehive patch requires the connect string of the database instance that is hosted on the computer that contains the Oracle Data Pump directories. In this connect string, use the `INSTANCE_NAME` parameter to specify which database instance contains the Oracle Data Pump directories. For example, the following specifies the database instance `afserv1` on node `host1.example.com`:

```
(DESCRIPTION=
  (ADDRESS_LIST=
    (ADDRESS= (PROTOCOL=TCP) (HOST=host1.example.com) (PORT=1521) ) )
  (CONNECT_DATA=
    (SERVER=DEDICATED)
    (SERVICE_NAME=afserv1.example.com)
    (INSTANCE_NAME=afserv1) )
```

Refer to the documentation of the Oracle Beehive patch for directions on how to specify the connect string. Refer to "Local Naming Parameters (tnsnames.ora)" in *Oracle Database Net Services Reference* and "Understanding the Oracle Real Application Clusters Installed Configuration" in *Oracle Real Application Clusters Installation Guide for Linux and UNIX* for more information about the `INSTANCE_NAME` parameter.

- Create the Oracle Data Pump directories in a Direct Network File System (NFS), then perform step 2 as described in ["Defining Oracle Data Pump Directories for Non-Oracle RAC Deployments"](#).

Backing up and Deleting Oracle Data Pump Log Files If you previously upgraded or patched Oracle Beehive, backup and delete Oracle Data Pump data and log files.

Disable User Directory Services Synchronization

This is a recommended but not mandatory step.

From any application tier, run the following command to disable User Directory Services (UDS) synchronization:

```
beectl modify_property --component OID_Profile --name ProfileState --value DISABLE
--activate_configuration
```

Shut down All Oracle Beehive Instances

Shut down all your Oracle Beehive application tiers by following these steps for each application tier:

1. Ensure that all Oracle Beehive processes are running with the following command:

```
beectl start --all
```

2. Run the following command repeatedly until the output indicates that all Oracle Beehive processes are running:

```
beectl status
```

3. Stop the application tier with the following command:

```
beectl stop --all
```

Note: Ensuring that all Oracle Beehive processes are running before shutting down the application tier ensures that any processes managed by OPMN remain stopped during upgrade.

If a computer goes down while OPMN is running, upon restart, OPMN will attempt to automatically restart all processes that were running at the time the system went down.

Consequently, upgrading an Oracle Beehive application tier that was shut down unexpectedly (for example, by rebooting the computer without first shutting down Oracle Beehive) may fail. When the upgrade process starts OPMN, OPMN will attempt to restart any of processes that were running, which in turn will cause the upgrade process to fail.

Upgrade Sequence

To upgrade an Oracle Beehive deployment, upgrade the following Oracle Beehive products in the indicated order:

1. Oracle Beehive. Refer to ["Oracle Beehive Upgrade Process Sequence of Screens"](#).

Note: You must shut down all Oracle Beehive application tiers as described in ["Shut down All Oracle Beehive Instances"](#) before upgrading them.

Ensure that the upgrade process has started your newly upgraded Oracle Beehive application tiers before proceeding to upgrade other Oracle Beehive products.

Refer to ["Upgrading Multiple Oracle Beehive Application Tiers"](#) if you are upgrading more than one Oracle Beehive application tier.

2. Any standalone Oracle Beehive Integration for Zimbra, Oracle Beehive Provisioning Application, or Oracle Collaboration Coexistence Gateway (Windows only).
3. Any Oracle Beehive for DMZ instances. Refer to ["Oracle Beehive for DMZ Upgrade Process Sequence of Screens"](#).

Note: You must shut down all Oracle Beehive for DMZ instances before upgrading them.

Ensure that the upgrade process has started your newly upgraded Oracle Beehive DMZ instances before proceeding to upgrade other Oracle Beehive products.

4. Any Oracle Beekeeper Release instances, refer to ["Upgrading Oracle Beekeeper"](#).

Note: Before upgrading Oracle Beekeeper, refer to the section ["Upgrading Oracle Beekeeper Version 1.5.x to Version 2.0"](#).

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

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.

Upgrading Multiple Oracle Beehive Application Tiers

Follow these steps to upgrade multiple Oracle Beehive application tiers:

1. Shut down all your Oracle Beehive application tiers as described in ["Shut down All Oracle Beehive Instances"](#).
2. Upgrade an Oracle Beehive application tier.
3. Wait until the upgrade process is complete.
4. Upgrade a subsequent Oracle Beehive application tier. Do not start upgrading this tier until the previous tier's upgrade process is complete. You do not have to shut down any upgraded application tiers.
5. Repeat step 4 until all application tiers are upgraded.

Upgrading Oracle Beekeeper Version 1.5.x to Version 2.0

To upgrade Oracle Beekeeper version 1.5.x to version 2.0, follow these steps:

1. If you have configured Oracle Beekeeper for SSL access, follow these steps. Otherwise, proceed to the next step.

These steps involve reconfiguring Oracle Beekeeper for SSL with the default-web-site.xml file:

- a. Copy `<Oracle Beekeeper home>/j2ee/home/config/secure-web-site.xml` as `<Oracle Beekeeper home>/j2ee/home/config/default-web-site.xml` (replacing default-web-site.xml with secure-web-site.xml)

- b. Edit the file `<Oracle Beekeeper home>/j2ee/home/config/server.xml` and replace `secure-web-site.xml` with `default-web-site.xml`.
 - c. Restart Oracle Beekeeper with the following commands and verify that SSL is working properly:


```
<Oracle Beekeeper home>/opmn/bin/opmnctl stopall
<Oracle Beekeeper home>/opmn/bin/opmnctl startall
```
2. Determine if Oracle Beekeeper is using the database or your LDAP server for authentication. Open the file `<Oracle Beekeeper home>/j2ee/home/applications/javasso/jps-config.xml`. Search for the element `<jpsContexts>`. The value of the default attribute may be either `ldap` or `db`:
 - If it is `db`, then no further action is required; do not proceed with the following steps. Upgrade Oracle Beekeeper version 1.5.x to Oracle Beekeeper 2.0; refer to ["Upgrading Oracle Beekeeper"](#).
 - If it is `ldap`, then perform steps 3-6.
3. Save a copy of the following files:
 - `<Oracle Beekeeper version 1.5 home>/j2ee/home/application-deployments/javasso/jps-config.xml`
 - `<Oracle Beekeeper version 1.5 home>/j2ee/home/application-deployments/beehivecontrol/jps-config.xml`
4. Upgrade Oracle Beekeeper version 1.5.x to version 2.0; refer to ["Upgrading Oracle Beekeeper"](#).
5. To configure LDAP-based authentication for your upgraded Oracle Beekeeper version 2.0 instance, modify the following files with the data contained in the files you saved in step 3:
 - `<Oracle Beekeeper version 2.0 home>/j2ee/home/application-deployments/javasso/jps-config.xml`
 - `<Oracle Beekeeper version 2.0 home>/j2ee/home/application-deployments/beehivecontrol/jps-config.xml`

Refer to the section ["Configuring Oracle Beekeeper for LDAP-Based Authentication"](#) in ["Oracle Beekeeper Post-Installation Procedures"](#) for more information about which attribute values you must modify in these files to configure LDAP-based authentication.
6. Restart the Oracle Beekeeper unmanaged OC4J instance with the following commands:


```
<Oracle Beekeeper home>/opmn/bin/opmnctl stopall
<Oracle Beekeeper home>/opmn/bin/opmnctl startall
```

Post-Upgrade Procedures

The following topics are covered in this section:

- [Running Perl Script `post_upgrade_db_actions.pl`](#)
- [Applying Deployment Template after Upgrade](#)
- [Gathering Statistics About Oracle Beehive Data and Code Schemas After Upgrading](#)

- [Upgrading Voicemail Configuration](#)
- [Re-enabling UDS Synchronization](#)
- [Upgrading User Directory Services](#)
- [Changing Permissions of hasbind](#)
- [Upgrading Mobile Device Management Service After Oracle Beehive Upgrade](#)

Running Perl Script `post_upgrade_db_actions.pl`

Note: Run the script `<Oracle Beehive home>/beehive/db/post_upgrade_db_actions.pl` only if the following conditions are true:

- You are applying a patch to Oracle Beehive Release 2 (2.0) or later.
- The patch you are applying involves schema cloning.

In particular, run this script if you did **not** specify the option `-no_schema_cloning` when you applied the patch.

The `-no_schema_cloning` option of the `opatch` command line tool enables you to apply a patch without schema cloning, which may potentially reduce the time required to apply the patch. Only use this option if you can fully shut down processes in all your Oracle Beehive application tiers, including Oracle Beekeeper instances. Note that this option is not recommended because it disables ZDU.

Run the script `post_upgrade_db_actions.pl` as follows:

```
perl post_upgrade_db_actions.pl <BEE_DATA> <OLD_BEE_CODE> <NEW_BEE_CODE> <BEE_CODE_PASSWORD> <CONNECT_STRING>
```

- `<BEE_DATA>`: Name of the Oracle Beehive data schema
- `<OLD_BEE_CODE>`: Name of the old Oracle Beehive code schema
- `<NEW_BEE_CODE>`: Name of the cloned and upgraded Oracle Beehive code schema
- `<BEE_CODE_PASSWORD>`: Password for the Oracle Beehive schemas (all Oracle Beehive schemas have the same password)
- `<CONNECT_STRING>`: Oracle Beehive database connect string

Note: The database connect string cannot contain any new line characters (`\n`, `\r`, or `\n\r`); the connect string must be a single-line value.

To retrieve the names of these schemas, run the following command:

```
beectl list_schemas
--schema_type <schema type>
--status <schema status>
--sort_by <sort condition>
```

- `<schema type>`: Type of schema to retrieve; it may have one of the following values:

- 1: Code schema
- 2: Data schema
- 3: Search-related Change Data Capture publisher (CDCPUB) schema
- **<schema status>:** Status of the schema. For purposes of running the script `post_upgrade_db_actions.pl`, you use only statuses 4, 5, and 6:
 - 1: Created; the code schema has been newly created
 - 2: Upgrade ready; the code schema has been cloned and is ready to be upgraded
 - 3: Activation ready; the code schema has been upgraded and is ready to be activated
 - 4: Active; the schema is active
 - 5: Legacy; the original schema that was cloned is set to this status. During a multi-application tier upgrade, application tiers that have not been upgraded will use this schema.
 - 6: Deactivated; when all application tiers have been upgraded, the original schema is set to this status
 - 7: Deinstalled; the schema has been deinstalled

Running `beectl list_schemas` without any options lists all schemas.

Examples

The following examples show you how to retrieve the names of schemas required for the `post_upgrade_db_actions.pl` script.

- The following example lists all data schemas:

```
beectl list_schemas --schema_type 2
```

```
schema_name: BEE_DATA schema_id: 131 version_id: 2.0.1.0.0
schema_type: 2 status: 4 creation_time: 2009-05-02 11:29:54.0
activation_time: 2009-05-02 11:29:54.0 description: BEE_DATA schema
```

- The following example lists all code schemas:

```
beectl list_schemas --schema_type 1
```

```
schema_name: BEE_CODE schema_id: 132 version_id: 2.0.1.1.0
schema_type: 1 status: 6 creation_time: 2009-05-02 11:29:54.0
activation_time: 2009-05-02 11:29:54.0 legacy_time: 2009-05-04
09:30:07.0 deactivation_time: 2009-05-04 10:14:57.0 description: BEE_CODE
schema

schema_name: BEE_CODE_05042009 schema_id: 134 version_id: 2.0.1.1.0
schema_type: 1 status: 6 creation_time: 2009-05-04 09:00:30.0
activationready_time: 2009-05-04 09:29:20.0 upgradeready_time:
2009-05-04 09:28:43.0 activation_time: 2009-05-04 09:30:07.0
legacy_time: 2009-05-04 13:11:42.0 deactivation_time: 2009-05-11 12:06:02.0
description: insert description here

schema_name: BEE_CODE_05042009_1 schema_id: 135 version_id: 2.0.1.1.0
schema_type: 1 status: 4 creation_time: 2009-05-04
12:26:03.0 activationready_time: 2009-05-04 12:52:50.0
activation_time: 2009-05-04 13:11:42.0 description: insert description here
```

```
schema_name: BEE_CODE_05112009 schema_id: 136 version_id: 2.0.1.1.0
schema_type: 1 status: 2 creation_time: 2009-05-11 12:13:43.0
upgradeready_time: 2009-05-11 12:48:14.0 description: insert description here
```

- The following example lists the active code schema. At the end of an upgrade involving schema cloning, the active code schema should be the cloned schema:

```
beectl list_schemas --schema_type 1 --status 4
```

```
schema_name: BEE_CODE_05042009_1 schema_id: 135 version_id: 2.0.1.1.0
schema_type: 1 status: 4 creation_time: 2009-05-04
12:26:03.0 activationready_time: 2009-05-04 12:52:50.0
activation_time: 2009-05-04 13:11:42.0 description: insert description here
```

- The following example lists all code schemas that are marked "LEGACY" and the sorts them by the time they were marked this status. The newest schema in this list is the old code schema:

```
beectl list_schemas --schema_type 1 --status 5 --sort_by LEGACY_TIME
```

- The following example lists call deactivated code schemas. After running the `post_upgrade_db_actions.pl` script, the old code schema will be marked as deactivated.

```
beectl list_schemas --schema_type 1 --status 6
```

Applying Deployment Template after Upgrade

It is highly recommended that you apply a deployment template to your upgraded Oracle Beehive Release 2 (2.0) deployment provided that it is not already applied; the upgrade process does not automatically do this for you. Note that a new Oracle Beehive Release 2 (2.0) installation already has a deployment template associated with it.

A deployment template is an XML file that represents the formally defined structure of an Oracle Beehive application tier and its components such as OC4J instances, services, Oracle Beehive Transport Infrastructure (BTI), and the HTTP server.

If your upgraded Oracle Beehive Release 1 (1.4) deployment does not have a deployment template associated with it, those `beectl` commands that change the deployment structure (such as those that add and delete OC4J and service instances) will succeed. However, you will receive a message indicating that you should apply a deployment template.

In addition, future upgrades will fail if your deployment does not have a deployment template associated with it.

Follow these steps to apply a deployment template to an Oracle Beehive deployment:

1. Retrieve a list of available deployment templates with the command `beectl list_deployment_templates`. This command will output the identifier of each deployment template and a short description.
2. Select an appropriate deployment template and apply it with the `beectl modify_deployment_structure`. The following example applies the deployment template `SERVER_AND_CLIENT` to the local Oracle Beehive application tier:

```
modify_deployment_structure --primary_template SERVER_AND_CLIENT
```

Note: Any customization to the deployment structure (such as extra OC4J or service instances) or start/stop parameters (such as the maximum heap size of an OC4J instance) will be lost when you apply a deployment template with the command `beectl modify_deployment_structure`. The deployment template specified by this command will overwrite any customization in your Oracle Beehive deployment.

Gathering Statistics About Oracle Beehive Data and Code Schemas After Upgrading

After upgrading to Oracle Beehive Release 2 (2.0) and before your users access your upgrade Oracle Beehive deployment, you must gather statistics about the Oracle Beehive data and code schemas. Otherwise, you may experience serious performance degradation or service interruptions.

This section contains the following topics:

- [Gathering Statistics About Oracle Beehive Data Schema](#)
- [Gathering Statistics About Oracle Beehive Code Schema](#)

Refer to "Gathering Statistics with DBMS_STATS Procedures" in the chapter "Managing Optimized Statistics" in *Oracle Database Performance Tuning Guide* for more information.

Gathering Statistics About Oracle Beehive Data Schema

Run the following SQL*Plus command as the SYS user to gather statistics about the Oracle Beehive data schema:

```
SQL> exec DBMS_STATS.GATHER_SCHEMA_STATS('BEE_DATA');
```

BEE_DATA is the name of the Oracle Beehive data schema.

Gathering Statistics About Oracle Beehive Code Schema

Run the following SQL*Plus commands as the SYS user to gather statistics about the Oracle Beehive code schema:

```
SQL> exec DBMS_STATS.GATHER_SCHEMA_STATS('<CODE_SCHEMA>');
```

<CODE_SCHEMA> is the name of the code schema user. To retrieve the name of the code schema, run the `beectl list_schemas` command as follows:

```
beectl list_schemas --schema_type 1 --status 4
```

```
schema_name: BEE_CODE_05042009_1 schema_id: 135 version_id: 2.0.1.1.0
schema_type: 1 status: 4 creation_time: 2009-05-04
12:26:03.0 activationready_time: 2009-05-04 12:52:50.0
activation_time: 2009-05-04 13:11:42.0 description: insert description here
```

Refer to "[Running Perl Script post_upgrade_db_actions.pl](#)" for more information about the `beectl list_schemas` command.

Upgrading Voicemail Configuration

After you have upgraded Oracle Beehive from an earlier version to version 1.4, you must re-create your voicemail facilities using the new method, and remove facilities that you created in earlier version with the `beectl add_config_object` command.

You can list facilities and groups created using the earlier method by using the following command (from the operating system shell, so you can make use of the `grep` utility):

```
beectl list_components | grep -i voice
```

Locate all the voice components defined with a voicemail DNIS alias. Then, check which group is associated to each voicemail DNIS by using the `beectl list_properties` command:

```
beectl list_properties --component <voicemail DNIS>
```

Run this command for each identified component, and make a note of the group associated with each voicemail DNIS.

Once you have this information, you can assign the groups and phone numbers using the new facility configuration method, by using the new `beectl add_voice_facility` command. The following example demonstrates briefly how to use the command:

```
beectl add_voice_facility
--group_collabid <GROUP_COLLAB_ID>
--include "18885551111|18885552???"
--exclude "188855529???"
```

See Also: For complete information on creating voicemail facilities, see "Managing Oracle Beehive Voicemail and Fax" in the *Oracle Beehive Administrator's Guide*.

The `--include` statement associates phone number 18885551111 and phone number range 18885552000-18885552999.

The `--exclude` statement associates the phone number range 18885552900-18885552999 **not** to be included in the broader include range.

The value of `--group_collabid` is the CollabID of a group. You can find this value for any group by using the `beectl list_groups` command with the global option `--entity_format id`:

```
beectl list_groups --group <group identifier> --show ALL --entity_format id
```

Use this command with the group that was defined for voicemail. If you followed the upgrade procedure described earlier to gather all the information, then the value for the `<GROUP_COLLAB_ID>` was listed when you used the `beectl list_properties` command.

Note: If you upgrading from 1.5.x to 2.0.x, comment out the `vm_purge.sql` before executing the script `vm_seed.sql`.

Re-enabling UDS Synchronization

Note: This section only applies to patches for Oracle Beehive Release 2 (2.0).

If you disabled UDS synchronization as described in ["Disable User Directory Services Synchronization"](#), run the following command from any application tier to enable it:


```
beectl modify_property --component OID_Profile --name ProfileState --value ENABLE
--activate_configuration
```

Upgrading User Directory Services

After upgrading Oracle Beehive to version 2.0, run the following SQL scripts:

Note: The system should not be made accessible to users until all the following scripts have been completed successfully.

1. Run the following script as the code schema user of your upgraded Oracle Beehive instance.

```
Select count(*) from user_scheduler_jobs where job_name like 'PATCH_UDS_CON_
GUARD_UPD_%';
```

To retrieve the name of the code schema, refer to "[Gathering Statistics About Oracle Beehive Code Schema](#)". If you see a count greater than 0, this means that the contact sensitivity upgrade process is still running. Rerun this script to check if the contact sensitivity fixing dbms jobs have completed. If you see a count equal to 0, this means that the contact sensitivity upgrade process is over. Please execute the next script.

2. Run the following SQL as the code schema user of your upgraded Oracle Beehive instance.

```
Select job_name, status, additional_info from user_scheduler_job_run_details
where job_name like 'PATCH_UDS_CON_GUARD_UPD_%';
```

If the status of any of the job is not 'SUCCEEDED', that means the contact sensitivity fixing dbms jobs have not successfully completed. Please contact support.

If the status for all the jobs is 'SUCCEEDED', it means that the contact sensitivity fixing dbms jobs have successfully completed. Please execute the next script.

3. Apply the patch for defect 9854302 before proceeding to the next step. If this patch is not applied, then the post upgrade step may delete E-mail preferences which may result in users losing their Zimbra e-mail preferences.
4. Run the following script as the BEE_DATA user, which is the name of the Oracle Beehive data schema.

```
<Oracle Beehive home>/beehive/tmp/patch/7685703/db/uds_post_upgrade_data.sql;
```

Run this script only after the previous script has been successfully executed. On successful execution of this script, run the next script.

5. Run this script as the code schema user of your upgraded Oracle Beehive instance:

```
<Oracle Beehive home>/beehive/tmp/patch/7685703/db/uds_post_upgrade_seed.sql;
```

If you see the message "Contact sensitivity fixing dbms jobs are running. Please execute the script after some time" this means that the contact sensitivity upgrade process is still running. Rerun the script at a later time. Run this script only after the previous scripts have been successfully completed.

This script will start a job to cleanup entries. You can now proceed with the execution of the subsequent steps. However no further patch should be applied till the clean job has been completed.

6. To check the status of the job, run the following query:

```
SQL> SELECT COUNT(*) FROM user_scheduler_jobs WHERE job_name LIKE 'PATCH_UDS_
POST_UPG_2_0_JOB%';
```

Run this script as the code schema user of your upgraded Oracle Beehive instance. If you see a count greater than 0, this means that the cleanup process is still running. No further patches should be applied. Rerun this script at a later time to check if the cleanup dbms jobs have completed. If you see a count equal to 0, it means that the cleanup dbms job is over.

7. Execute the next script.

```
SQL> SELECT job_name, status, additional_info FROM user_scheduler_job_run_
details WHERE job_name LIKE 'PATCH_UDS_POST_UPG_2_0_JOB%';
```

Run this script as the code schema user of your upgraded Oracle Beehive instance. If the status of the job is not 'SUCCEEDED', it means that the cleanup dbms job has not successfully completed.

If the status for all the jobs is 'SUCCEEDED', it means that the cleanup dbms jobs have successfully completed.

Changing Permissions of hasbind

If you have changed the permissions of hasbind as described in the section ["Changing Other Ports"](#) in ["Upgrading Oracle Beehive Overview"](#), then you must change the permissions of hasbind again after you have upgraded Oracle Beehive.

Upgrading Mobile Device Management Service After Oracle Beehive Upgrade

If you configured Oracle Application Server Single Sign-On for your Oracle Beehive Release 1 (1.5.x) deployment, and you upgraded it to Oracle Beehive Release 2 (2.0), follow these steps to upgrade the Mobile Device Management Service:

1. List all service instances of type MobileDmServiceInstance:

```
beectl list_components --type MobileDmServiceInstance
```

Component type	Component identifier
MobileDmServiceInstance	<instance_####1>
MobileDmServiceInstance	<instance_####2>

2. Upgrade each service instance of type MobileDmServiceInstance. For example:

```
beectl> upgrade_service_instance --service_instance_id <instance_####1>
beectl> upgrade_service_instance --service_instance_id <instance_####2>
```

3. Restart all BEEAPP and HTTP_Server components:

```
beectl status
```

Component identifier	Component type	Status
BTI_instance1.example.com	BTI	RUNNING
BEEAPP_instance1.example.com	ManagedOc4j	RUNNING
BEEMGMT_instance1.example.com	ManagedOc4j	RUNNING

```

-----+-----+-----
BEECORE_instance1.example.com | ManagedOc4j | RUNNING
-----+-----+-----
oc4j_soa_instance1.example.com | ManagedOc4j | RUNNING
-----+-----+-----
ohs_instance1.example.com | HttpServer | RUNNING
-----+-----+-----

```

```

beectl restart --component BEEAPP_instance1.example.com
beectl restart --component ohs_instance1.example.com

```

Upgrading Oracle Beehive by Clone Patching

Clone patching is an alternative way to upgrade an Oracle Beehive application tier. It involves cloning an already upgraded application tier, then replacing an old application tier with the clone. Clone patching is recommended if you have many application tiers to upgrade.

Refer to ["Oracle Beehive By Clone Patching"](#) for more information.

Creating Batches For Use With add_search_recovery_scope Command

After upgrading to Beehive versions later than 2.0, use the following steps to identify and create batches for use with the add_search_recovery_scope command to build a new search index.

Recovery with the add_search_recovery_scope command should be run during off-hours and with limited sets of data. Ensure that recovery is not being done when there is a scheduled service bounce as having BEEAPP bounce while running recovery will cause recovery to hang. Ensure that there is no loss of connectivity to the database and recovery is not run during scheduled maintenance.

The add_search_recovery_scope command only needs to be run to index old content. New e-mails and new /modified calendar events will automatically be indexed if the CaptureChanges value is set to true for the _SearchService component. This can be verified with the following command:

```
beectl list_properties --component _SearchService --name CaptureChanges
```

and, if found to be false, can be set back to true with the following command:

```
beectl modify_property --component _SearchService --name CaptureChanges --value true --activate_configuration
```

Follow the instructions below to separate workspaces into batches, recover the workspaces in each batch, and check statuses:

1. Getting workspaces:

```
beectl list_workspaces --scope enpr=EnterpriseName --recurse --entity_format id
--format tiled --countLimit 100000
```

The above command returns an entry for each workspace with the following data:

```
Name of workspace | type of workspace | Workspace ID | Enterprise ID
```

For example:

```
test2|TEAM|0B69:7BD3:wstm:6F933CCBA840488E9C3BCE8318260F1F00000000010|0B69:7BD
3:enpr:82170304C0F3F514E040940A832C31CD000000018895|
user1.test's Personal
```

```
Workspace|PERSONAL|0B69:7BD3:wsp:9027544241E4497EA90D8F7DBD5896D5000000000001|
0B69:7BD3:enpr:82170304C0F3F514E040940A832C31CD000000018895|
```

Column 3 contains the workspace collab IDs to be used in the next step.

2. Compiling beectl commands to recover workspaces:

From output of step 1, create a file called `batch.txt` to contain the `add_search_recovery_scope` command defined for the workspaces and date range as shown in the following example. The number of workspaces and/or date range will depend on the average content size of your workspaces, the batch size can be modified. The recommended batch size is 10,000 - 20,000 items, it is further recommended that no more than 50,000 items are submitted and/or are processed at any given time.

```
add_search_recovery_scope --scope
0B69:7BD3:wstm:6F933CCBA840488E9C3BCE8318260F1F000000000010 --start_date
2009-03-01T00:00:00 --end_date 2010-10-31T00:00:00
```

```
add_search_recovery_scope --scope
0B69:7BD3:wsp:9027544241E4497EA90D8F7DBD5896D5000000000001 --start_date
2009-03-01T00:00:00 --end_date 2010-10-31T00:00:00
```

Note: If `start_date` and `end_date` are not specified, by default only past 30 days content will be recovered.

3. Running recovery commands: The command will recover the workspaces in the batch file.

```
beectl --batchfile batch.txt
```

For example:

```
*****
Batch file execution output
Batch file name : /home/test1/batch.txt
Execution time : 10/6/10 7:51 AM
*****
Command number : 1
Command line : add_search_recovery_scope --scope
0B69:7BD3:wstm:6F933CCBA840488E9C3BCE8318260F1F000000000010
Command output is shown below :
Search index recovery started. Please use the following beectl command to
monitor its progress.
list_operation_statuses --operation_status
12A6:2F23:opst:91BB7AE87FEDE688E040E80A5E4B368900000004E1BA
Command exit code : 0
*****
Command number : 2
Command line : add_search_recovery_scope --scope
0B69:7BD3:wsp:9027544241E4497EA90D8F7DBD5896D5000000000001
Command output is shown below :
Search index recovery started. Please use the following beectl command to
monitor its progress.
list_operation_statuses --operation_status
12A6:2F23:opst:91BB7AE87FEDE688E040E80A5E4B368900000004E24B
Command exit code : 0
```

4. Monitoring operation status:

Use `list_operation_statuses` to monitor each recovery submission (as in step 3).

Similarly, monitoring operation status can be done in a batch as follows. Create `batch_status.txt` to contain the `list_operations_statuses` commands:

```
list_operation_statuses --operation_status
12A6:2F23:opst:91BB7AE87FEDE688E040E80A5E4B368900000004E1BA --format tiled

list_operation_statuses --operation_status
12A6:2F23:opst:91BB7AE87FEDE688E040E80A5E4B368900000004E24B --format tiled
```

Then run the following command to check the statuses:

```
beectl --batchfile batch_status.txt
```

If certain workspace recovery status is not completed and is not changing for a long time, record those workspaces and continue.

5. Monitoring database

The `add_search_recovery_scope` command adds feeds to the database. After a while the number of feeds should be stable. Some database jobs consume feeds to produce crawled entities. When `is_indexed` is 1, the entry is indexed.

The following two methods can be used to check feeds and crawled entities:

- `use ss_utils_pkg`

login to database as active BEE_CODE user (BEE_CODE_XXXX):

```
set serveroutput on;
declare
begin
ss_utils_pkg.searchstate();
end;
/
```

- `query ss_feeds, and ss_crawled_entities directly`

login to the database as active BEE_CODE user (BEE_CODE_XXXX):

```
SQL> select count(*) from ss_feeds;
```

this will tell the number of feeds. The value should be stable.

```
SQL> select count(*) from ss_crawled_entities where is_indexed in (0, -1, 2);
```

this will tell the number of crawled entities under indexing.

Wait for a while until both the numbers of feeds and of `ss_crawled_entities` are stable before handling the next batch.

6. Continue from step 2 to handle the next batch.

Upgrading Oracle Beehive

This chapter describes how to upgrade the following products to version 2.0:

- Oracle Beehive
- Oracle Beehive Provisioning Application version 1.5.x and later
- Oracle Beehive for DMZ version 1.5.x and later

Note: Upgrade Oracle Beehive and any other Oracle Beehive application tiers before upgrading any other Oracle Beehive products.

To upgrade an Oracle Beehive Release 1 (1.5.x) deployment to Release 2 (2.0), upgrade the following Oracle Beehive products in the indicated order:

1. Oracle Beehive Release 1 (1.5.x) application tiers to Release 2 (2.0).
You must shut down all Oracle Beehive Release 1 (1.5.x) application tiers before upgrading them.

Ensure that the upgrade process has started your newly upgraded Oracle Beehive application tiers before proceeding to upgrade other Oracle Beehive products.

Refer to "[Upgrading Multiple Oracle Beehive Application Tiers](#)" if you are upgrading more than one Oracle Beehive application tier.
 2. Any Oracle Beehive Provisioning Application or Oracle Collaboration Coexistence Gateway (Microsoft Windows only) Release 1 (1.5.x) instances to Release 2 (2.0).
 3. Any Oracle Beehive Release 1 (1.5.x) DMZ instances to Release 2 (2.0).
You must shut down all Oracle Beehive Release 1 (1.5.x) DMZ instances before upgrading them.

Ensure that the upgrade process has started your newly upgraded Oracle Beehive DMZ instances before proceeding to upgrade other Oracle Beehive products.
 4. Any Oracle Beekeeper Release 1 (1.5.x) instances to Release 2 (2.0).
-

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 computer 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 (1.5.x) instances on the computer from which you started the wizard, the following screens will appear:

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

Note: You must shut down all Oracle Beehive Release 1 (1.5.x) application tiers before upgrading them.

Ensure that the upgrade process has started your newly upgraded Oracle Beehive application tiers before proceeding to upgrade other Oracle Beehive products.

Refer to "[Upgrading Multiple Oracle Beehive Application Tiers](#)" if you are upgrading more than one Oracle Beehive application tier.

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 (1.5.x) instances, specify **Yes** to upgrade one of those instances to Oracle Beehive Release 2 (2.0).

If you specify **No**, the Install Wizard will proceed to perform a standard installation of Oracle Beehive Release 2 (2.0).

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 (1.5.x) or Oracle Beehive Integration for Zimbra version 1.5.x
- Is registered in the central Oracle Universal Installer inventory of the computer from which you started the Install Wizard.

Running 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.

You must manually verify that all unmanaged instances (listed in red) are shut down. Unmanaged instances include Oracle Beekeeper instances and Oracle Beehive DMZ instances. Currently, Oracle Beehive cannot determine whether an unmanaged instance has been fully shut down (regardless of whether the instance is on the same machine as the pending upgrade or not). Once you have shut down all unmanaged instances, select **Unmanaged Instances Verified** to continue the upgrade.

Configure Security Updates

Specify your My Oracle Support account details so that Oracle can notify you of any critical security updates.

Periodically, Security Updates automatically gathers configuration information of your installed Oracle products and uploads it to Oracle's support systems. Consequently, you may access this information through your My Oracle Support account and Oracle can contact you if there are any security updates.

Note: The information collected by Security Updates is limited to configuration information. The data collected does not include personally identifiable information (with the exception of a local contact name in case of transmission problems). You may still use all licensed Oracle functionality if you decline to enable Security Updates.

You may choose not to be notified for any critical security updates. Simply leave all fields in the Configure Security Updates screen blank and click **Next** to continue.

E-mail

Specify your My Oracle Support e-mail address. If you do not have a My Oracle Support account, you may specify a personal e-mail address.

I wish to receive security updates via My Oracle Support

Select this check box if you want to receive security updates through My Oracle Support. Ensure that the e-mail address you entered in **Email** and the password you entered in **My Oracle Support Password** corresponds to your My Oracle Support user name and password, respectively.

If you specified a personal e-mail address in **Email**, then ensure that this check box is not selected.

My Oracle Support Password

Specify your My Oracle Support password.

Specify proxy server information

If you click **Next** and the Install Wizard cannot establish a direct connection to an outside network to send your My Oracle Support information, the **Specify proxy server information** window appears. Enter the following information:

- **Proxy Server:** The host name of your proxy server
- **Proxy Port:** The port number of your proxy server
- **Proxy Username:** The user name required to authenticate your proxy server, if required
- **Proxy Password:** The password required to authenticate your proxy server, if required
- **I want to remain uninformed of critical issues in my configuration:** Selecting this check box disables Security Updates. If you select this check box, you do not need to specify any proxy details.

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.

Note: 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.
 - **Running Instances:** Oracle Beehive instances that are still running. You must shut down and then verify that all Oracle Beehive instances are down before proceeding with the upgrade.

- **Database Information for Upgrade:** The masked password of the SYS schema of the existing Oracle Database instance that will be used by this upgrade process.
- **Site Key:** The masked site key that you specified. This is only requested if you are upgrading your first Oracle Beehive application tier.

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:

- Upgrade Pre-Configuration
- Upgrading Central Inventory
- Enabling Home Instance
- Configuring OPMN Memory Parameters (this is performed only if the Oracle home being upgrade is a standalone Oracle Beehive Integration for Zimbra home)
- Base Platform Configuration
- Stopping OPMN
- Upgrading SOA Schemas (this is performed only for the first Oracle Beehive application tier being upgraded)
- Preparing OWSM Upgrade
- Disabling Beehive and Home Instances
- Starting OPMN
- Performing OWSM Upgrade
- Performing ESB Upgrade
- Stopping OPMN
- Running OPMN Log Configuration Assistant
- Finalizing Application Server 10.1.3.4.0 Updates
- Upgrading Beehive Schemas
- Upgrading Beehive Midtier
- Performing Post-Upgrade Actions

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 1.5.x is installed on the computer 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 1.5.x is installed in your computer, specify **Yes** to upgrade it to version 2.0.

If you specify **No**, the Install Wizard will proceed to install Oracle Beehive Provisioning Application version 2.0 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 1.5.x
- Is registered in the central Oracle Universal Installer inventory of the computer 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.

Note: 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 2.0, a component named after the version number to which you upgraded will be added to the Oracle Enterprise Manager Grid Control software library.

Oracle Beehive for DMZ Upgrade Process Sequence of Screens

If the Oracle Beehive Wizard detects that Oracle Beehive for DMZ version 1.5.x is installed on the computer 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](#)

Note: You must shut down all Oracle Beehive Release 1 (1.5.x) DMZ instances before upgrading them.

Ensure that the upgrade process has started your newly upgraded Oracle Beehive DMZ instances before proceeding to upgrade other Oracle Beehive products.

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 1.5.x is installed in your computer, specify **Yes** to upgrade it to version 2.0.

If you specify **No**, the Install Wizard will proceed to install Oracle Beehive for DMZ version 2.0 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.
- Is registered in the central Oracle Universal Installer inventory of the computer 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.

Note: 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.

Upgrading Oracle Beekeeper

This chapter describes how to upgrade Oracle Beekeeper version 1.5.x and later to Oracle Beekeeper version 2.0.

Start the Oracle Beekeeper Install Wizard. If the Install Wizard detects that Oracle Beekeeper version 1.5.x or later is installed in your computer, you will see the screen [Upgrade Existing Home](#). Refer to "[Oracle Beekeeper Upgrade Process Sequence of Screens](#)" for a summary of the upgrade process.

Oracle Beekeeper Upgrade Process Sequence of Screens

If the Oracle Beekeeper Install Wizard detects that Oracle Beekeeper version 1.5 or later is installed on the computer from which you started the wizard, the following screens will appear:

- [Upgrade Existing Home](#)
- [Location to Upgrade](#)
- [Prerequisite Checks](#)
- [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 Install Wizard detected that Oracle Beekeeper version 1.5.x is installed in your computer, specify **Yes** to upgrade it to Oracle Beekeeper version 2.0.

If you specify **No**, the Install Wizard will proceed to install Oracle Beekeeper version 2.0 in a new Oracle 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 Beekeeper version 1.5.x
- Is registered in the central Oracle Universal Installer inventory of the computer from which you started the Install Wizard

Prerequisite Checks

This screen displays the prerequisite checks that the Install Wizard runs. It verifies that the host (where you are upgrading to Oracle Beekeeper) meets all minimum requirements.

Some of the platform-independent checks that the 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**.

Note: 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 for Upgrade

Specify the password of the BEE_CODE 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.

Note: 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 Beekeeper
- **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 Beekeeper.

The Oracle Beehive Install Wizard performs the following configuration assistants:

- Oracle Beekeeper Initialization
- Installing Application Server 10.1.3.4.0 Patchset
- Upgrading Required Applications
- Deconfiguring Oracle Beekeeper Instance
- Oracle Beekeeper Service Creation
- Starting OC4J Instance
- Oracle Beekeeper Application Deployment
- Configuring OC4J Application Settings
- Stopping OC4J Instance
- Starting OC4J Instance
- Performing Post-Upgrade Actions

End of Upgrade

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

Oracle Beehive By Clone Patching

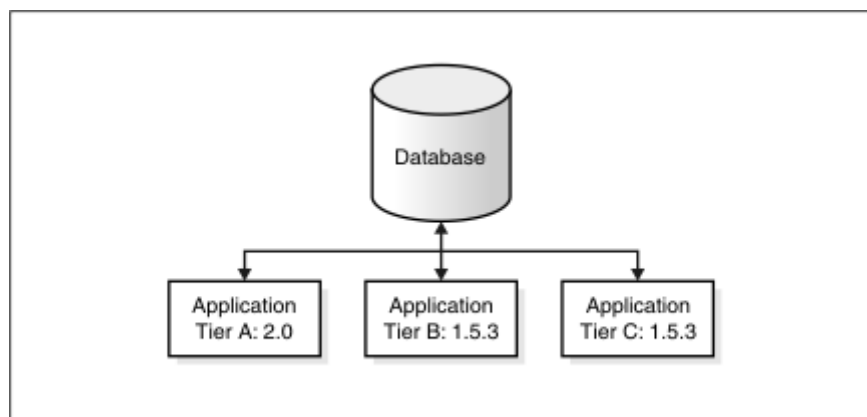
Clone patching is an alternative way to upgrade an Oracle Beehive application tier. It involves cloning an already upgraded application tier, then replacing an old application tier with the clone. Clone patching is recommended if you have many application tiers to upgrade.

Introduction to Clone Patching

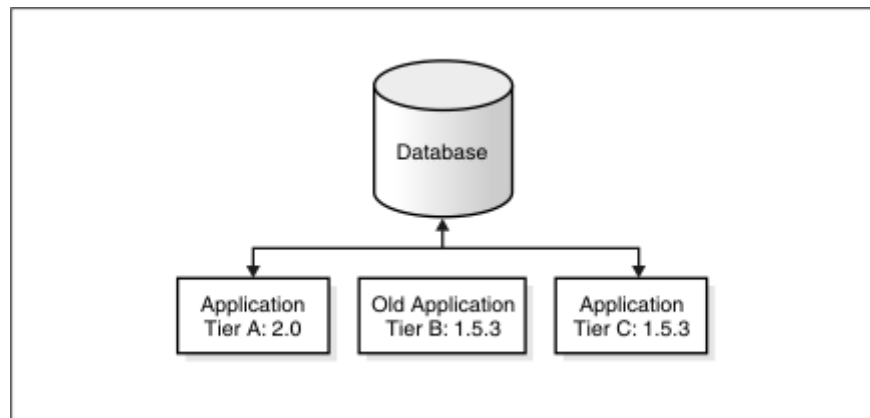
The following figures illustrate how clone patching works:

[Figure 14–1, "Oracle Beehive Deployment Before Clone Patching"](#) depicts an Oracle Beehive deployment with three application tiers - application tier A has been upgraded to Release 2 (2.0) and application tiers B and C are Release 1 (1.5.3).

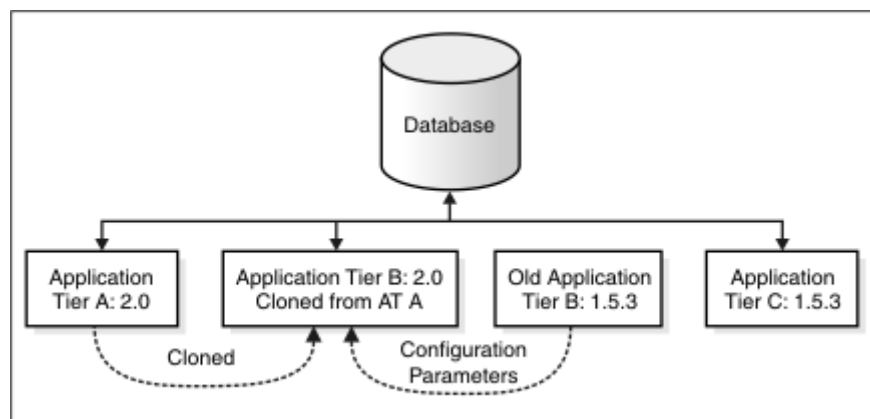
Figure 14–1 Oracle Beehive Deployment Before Clone Patching



In [Figure 14–2, "Backing up Application Tier Before Clone Patching"](#), application tier B has been shut down and backed up.

Figure 14–2 Backing up Application Tier Before Clone Patching

In [Figure 14–3, "Oracle Beehive Deployment After Clone Patching"](#), application tier B has been upgraded to Release 2 (2.0) from clone patching application tier A. Some configuration parameters have been copied from the backed up application tier B to the upgraded application tier B.

Figure 14–3 Oracle Beehive Deployment After Clone Patching

Types of Clone Patching

You may upgrade Oracle Beehive by clone patching in the following two ways:

Specifying the location of the backed up Oracle Beehive home

This is the recommended method of upgrading Oracle Beehive by clone patching. This method involves creating a backup of the Oracle Beehive instance you want to upgrade by moving it, not copying it, to a new directory. When you run the command `beectl clone_patching`, you specify the location of the backed up Oracle Beehive home with the `--old_oracle_home` option.

An advantage of this method is that after clone patching, the application tier remains SSL or AJP enabled if it was SSL or AJP enabled prior to clone patching. It does not matter if the cloned, upgraded application tier is SSL or AJP enabled.

The following table describes the options for the `beectl clone_patching` command for this method:

Table 14–1 *beectl clone_patching Options for Specifying Backed up Oracle home*

Option	Mandatory/ Optional	Description
--old_oracle_home	Mandatory	Full path name of the location of the old, backed up Oracle Beehive home to be upgraded.
--oui_inv_ptr_loc	Optional Note: This option is available only for UNIX-based systems. Do not specify if it does not exist on the computer from which you are running this command; in this case, the Oracle inventory will be created in the user' home directory.	Note: This option is available only for UNIX-based systems. 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: --oui_inv_ptr_loc "/etc/oraInst.loc" Note: This value is platform-dependent. On Linux, it is /etc/oraInst.loc.
--start_at_end	Optional	If true, Oracle Beehive will start components after cloning. Permitted value is a boolean value. By default, components are not started at the end of clone patching.

Specifying the Oracle Beehive instance system model

In certain cases, you may not be able to specify the location of the old Oracle Beehive home file system. In such cases, when you run the command `beectl clone_patching`, use the option `--local_beehive_instance_id` (and related options `--site_key` and `--db_schema_password`).

A disadvantage to this method is that the upgraded application tier is not SSL or AJPS enabled even if it was prior to clone patching.

The following table describes the options for the `beectl clone_patching` command for this method:

Table 14–2 *beectl clone_patching Options for Specifying System Model Identifier*

Option	Mandatory/ Optional	Description
--local_beehive_instance_id	Mandatory	Target Oracle Beehive instance system model identifier. To get the value for Beehive instance ID, run the following <code>beectl</code> command: ./beectl list_components --type BeehiveInstance

Table 14–2 (Cont.) beectl clone_patching Options for Specifying System Model Identifier

Option	Mandatory/ Optional	Description
--site_key	Mandatory	<p>Site key of the site you are cloning. This is an alphanumeric string.</p> <p>Notes:</p> <ul style="list-style-type: none"> The site key you specify must be the same as the source site. If you are not in shell mode, you must obfuscate the site key and add the --obfuscated option to the beectl clone_patching command. <p>To obfuscate a password, use the beectl obfuscate command:</p> <pre>beectl obfuscate --expiration_time_in_minutes 0 Enter value for password:</pre> <p>Successfully obfuscated the string.</p>
--db_schema_password	Mandatory	<p>Database password for the schema.</p> <p>Note: If you are not in shell mode, you must obfuscate the database password and add the --obfuscated option to the beectl clone_patching command.</p> <p>To obfuscate a password, use the beectl obfuscate command:</p> <pre>beectl obfuscate --expiration_time_in_minutes 0 Enter value for password:</pre> <p>Successfully obfuscated the string.</p>
--oui_inv_ptr_loc	Optional	<p>Note: This option is available only for UNIX-based systems.</p> <p>Oracle Universal Installer inventory location.</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 "/etc/oraInst.loc"</pre> <p>Note: This value is platform-dependent. On Linux, it is /etc/oraInst.loc.</p>
--start_at_end	Optional	<p>If true, Oracle Beehive will start components after cloning. Permitted value is a boolean value.</p> <p>By default, components are not started at the end of clone patching.</p>

Clone Patching

Follow the below mentioned steps to upgrade an Oracle Beehive by clone patching:

Step 1 Verify Requirements

Ensure that the chipset and the operating system version (including any operating system patches) of all the application tiers you want to upgrade by clone patching is the same.

Step 2 Upgrade one of your application tiers

Refer to ["Upgrading Oracle Beehive Overview"](#) and ["Upgrading Oracle Beehive"](#) for more information.

Step 3 Create a source image of the upgraded application tier

Create a source image of the upgraded application tier by following the steps described in the section ["Preparing Source Application Tier Instance"](#) in ["Cloning Oracle Beehive Application Tiers and Sites"](#).

Step 4 Shut down the application tier you want to upgrade

Run the command `beectl stop --all` on the application tier you want to upgrade.

Step 5 Move the application tier you want to upgrade

Move the contents of the Oracle Beehive home you are upgrading to another directory:

```
mv $ORACLE_HOME $BACKUP_ORACLE_HOME
```

`$ORACLE_HOME` is the Oracle home of the application tier you are moving, and `$BACKUP_ORACLE_HOME` contains the contents of `$ORACLE_HOME`.

Step 6 Unzip the source image of the upgraded application tier

Unzip the source image of the upgraded application tier (which you created in step 3) in the same location as the Oracle Beehive home you are upgrading. (The Oracle home for this unzipped source image will be `$ORACLE_HOME`).

To unzip this source image, follow ["Step 1: Unzip Compressed Oracle Home"](#) in the section ["Application Tier Cloning"](#) in ["Oracle Beehive By Clone Patching"](#).

Step 7 Update ORACLE_HOME in the beectl command

Update `ORACLE_HOME` in the `beectl` command to refer to the upgraded target Oracle Beehive application tier as described in ["Step 3: Modify Oracle Home Path"](#) in ["Cloning Oracle Beehive Application Tiers and Sites"](#).

Step 8 Run the clone patching command

If you are using the clone method ["Specifying the location of the backed up Oracle Beehive home"](#), run the `beectl clone_patching` command as follows:

```
beectl clone_patching --old_oracle_home $BACKUP_ORACLE_HOME
```

If you are using the clone method ["Specifying the Oracle Beehive instance system model"](#), run the `beectl clone_patching` command as follows:

```
beectl clone_patching --site_key <secure site key> --db_schema_password <secure database password> --local_beehive_instance_id <Oracle Beehive instance system model ID>
```

The following is an example of running this command:

```
beectl clone_patching
--local_beehive_instance_id beehive_instance_instM2.example.com
--db_schema_password c2uf0aQ9IYRPPVFnsUOmng8BD3S+042hZxWlUYHn1jZx5EqOH5TkCA==
--site_key c2uf0aQ9IYRPPVFnsUOmng8BD3S+042hZxWlUYHn1jZx5EqOH5TkCA==
```

--obfuscated

Note: When using obfuscated passwords with the `beectl clone_patching` command, ensure the following:

- If your Oracle Beehive deployment consists of multiple application tiers, obfuscate passwords on the same application tier you plan to use them; do not use an obfuscated password created on one instance on another instance.
 - Do not enclose obfuscated passwords in single or double quotes.
-
-

Part III

Oracle Beehive Post-Installation Configuration

The Part III of the Installation Guide describes how to configure Oracle Beehive. In particular, it describes how to configure security-related issues, install Oracle Beehive Extensions for Outlook, and create Oracle Beehive clones ideal for testing changes to an Oracle Beehive instance before applying them to a production modules.

This part contains the following chapters:

- [Oracle Beehive Post-Installation Procedures](#)
- [Oracle Beehive Database Post-Installation Procedures](#)
- [Configuring FTP](#)
- [Configuring Oracle Beehive Demilitarized Zone Instances](#)
- [Configuring SSL](#)
- [Configuring TLS with Oracle Wallet](#)
- [Configuring SSL for LDAP Integration](#)
- [Enabling AJPS](#)
- [Configuring E-Mail with SSL](#)
- [Configuring XMPP](#)
- [Configuring Oracle Application Server Single Sign-On with Oracle Beehive](#)
- [Installing Oracle Beehive Extensions for Outlook](#)
- [Installing Oracle Beehive Extensions for Explorer](#)
- [Installing Oracle Beehive Conferencing Client](#)
- [Configuring Oracle Beehive Integration for Zimbra](#)
- [Cloning Oracle Beehive Application Tiers and Sites](#)
- [Oracle Beekeeper 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](#)
- [Performing Post-Installation Procedures on Oracle Beehive Database](#)
- [Configuring FTP](#)
- [Configuring Oracle Beehive to Listen on Ports Less Than 1024](#)
- [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 XMPP](#)
- [Configuring Actionable Notifications](#)
- [Creating ASK User](#)
- [Configuring Notifications to Use SMS](#)
- [Configuring Oracle Secure Enterprise Search](#)
- [Configuring Oracle Application Server Single Sign-On](#)
- [Installing Oracle Beehive Extensions for Outlook](#)
- [Installing Oracle Beehive Extensions for Explorer](#)
- [Installing Oracle Beehive Conferencing Client](#)
- [Configuring Oracle Beehive Integration for Zimbra](#)
- [Configuring Remote Voice Conferencing Media Server for Oracle Beehive Conferencing](#)
- [Cloning Oracle Beehive](#)
- [Performing Oracle Beekeeper 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*.

Performing Post-Installation Procedures on Oracle Beehive Database

After successfully installing Oracle Beehive, refer to "[Oracle Beehive Database Post-Installation Procedures](#)" for tuning recommendations to perform on your Oracle Beehive database.

Configuring FTP

Follow the steps described in "[Configuring FTP](#)" to enable and configure FTP connections to your Oracle Beehive instances.

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](#)

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_sitel.example.com</column>
  <column name="Property Name">HttpListenPort</column>
  <column name="Listening Component">ohs_sitel.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. If you have configured your Oracle Beehive deployment with Oracle Beehive DMZ instances, perform the following steps on all your DMZ instances:

- a. 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 <DMZ home>/beehive/bin/hasbind
chmod a+x <DMZ home>/beehive/bin/hasbind
chmod a+s <DMZ 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.

- b. As the root user, create a text file named /etc/cap.ora as described in Step 2 of this section.
 - c. Restart your Oracle Beehive DMZ instance.
4. 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`.

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

```
beectl modify_port --protocol BTP --port <new port number>
```

6. Change the appropriate property in the `_VIRTUAL_SERVER` component to the new port number. The following example changes the `BTiClientPort` 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>
```

7. 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.

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.

Oracle Beehive provides flexible user account management and provisioning by supporting both native and system-external user directory options. With Oracle Beehive, administrators can manage user account data either natively in Oracle Beehive or externally through integration with a supported LDAP-based user directory server. Oracle Beehive provides this flexibility for user account management through the User Directory Service.

Currently, Oracle Beehive supports the following user directory servers:

- Oracle Internet Directory
- IBM Tivoli Directory Server
- Microsoft Active Directory Server
- OpenLDAP Directory Server

- Oracle Directory Server Enterprise edition (formerly Sun Directory Server).

Note: If the external directory server is SunOne Enterprise Edition 7, ensure to enable the cn=changelog entry using "dsconf set-server-prop -h host -p port retro-cl-enabled:on" where host and port are the hostname of the directory server and port on which the server is listening.

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 an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide*.

Configuring SSL

Refer to "[Configuring SSL](#)" for information about configuring SSL with Oracle Beehive and DMZ instances.

If you do not want to use SSL with your Oracle Beehive deployment, follow the steps described in "[Installing Non-SSL Oracle Beehive Site](#)".

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 XMPP

Extensible Messaging and Presence Protocol (XMPP) is an open XML technology for presence and real-time communication. For users to authenticate against Oracle Beehive's XMPP Service, you must perform the steps described in "[Configuring XMPP](#)" to configure it.

Configuring Actionable Notifications

An actionable notification is an Oracle Beehive feature that allows users to accept or decline an invitation or assignment by replying to notifications they receive.

By default, however, notifications are one-way (from the server to the user). In order to enable actionable notifications, refer to "Configuring Actionable Notifications" in "Managing Oracle Beehive Subscriptions and Notifications" in *Oracle Beehive Administrator's Guide*.

Creating ASK User

When Actionable Notifications with SMS is enabled, users can interact with the ASK service interface, which allows them to send SMS commands to the Notification Delivery Service and retrieve e-mail, calendar, and contacts data.

For more information about the ASK service, refer to the *Oracle Beehive Registering and Configuring Mobile Devices* guide.

Oracle Beehive Mobile Communicator also relies on the ASK service.

To enable actionable notifications and the ASK service, an ASK user is required. For more information on setting up such a user, refer to "Configuring Actionable Notifications" in "Managing Oracle Beehive Subscriptions and Notifications" in *Oracle Beehive Administrator's Guide*.

Configuring Notifications to Use SMS

Oracle Beehive can send users notifications as e-mail messages, instant messages, or as mobile messages in the form of SMS. SMS notifications are only possible if the Oracle Beehive SMS delivery channel is enabled and configured to point to an SMS Aggregator.

For more information, refer to the section "Configuring Notifications to use SMS" in "Managing Oracle Beehive Subscriptions and Notifications" in *Oracle Beehive Administrator's Guide*.

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.

For more information, refer to "Integrating Oracle Secure Enterprise Search 10g with Oracle Beehive" in *Oracle Beehive Integration Guide*.

Configuring Oracle Application Server Single Sign-On

You may register Oracle Beehive as a partner application with Oracle Application Server Single Sign-On, which means that you may delegate the authentication function

to the single sign-on server. Refer to ["Configuring Oracle Application Server Single Sign-On with Oracle Beehive"](#).

Installing Oracle Beehive Extensions for Outlook

Oracle Beehive Extensions for Outlook 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.

Oracle Beehive Extensions for Outlook requires installation on the computers of individual users. For information about installing Oracle Beehive Extensions for Outlook, refer to ["Installing Oracle Beehive Extensions for Outlook"](#).

Installing Oracle Beehive Extensions for Explorer

Oracle Beehive Extensions for Explorer is an extension to Microsoft Windows Explorer that provides Oracle Beehive users direct access to their workspaces and workspace content, such as folders and documents. Oracle Beehive Extensions for Explorer also facilitates seamless team collaboration. For example, users can launch Oracle Beehive conferences directly from Windows Explorer, where they can share and discuss workspace content in real time.

Oracle Beehive Extensions for Explorer requires installation on the computers of individual users. For information about installing Oracle Beehive Extensions for Explorer, refer to ["Installing Oracle Beehive Extensions for Explorer"](#).

Installing Oracle Beehive Conferencing Client

The Oracle Beehive Conferencing client enables Oracle Beehive users to conduct Web-based meetings and presentations.

The Oracle Beehive Conferencing client requires installation on the computers of individual users. For information about installing the Oracle Beehive Conferencing client, refer to ["Installing Oracle Beehive Conferencing Client"](#).

Configuring Oracle Beehive Integration for Zimbra

Oracle Beehive Integration for Zimbra is installed and configured in the same Oracle home as Oracle Beehive if you install Oracle Beehive on a computer with at least 3 gigabytes (GBs) of memory and you select the Server + Client template during the installation process. If you have installed Oracle Beehive Integration for Zimbra in a different Oracle home than your Oracle Beehive instance, then it 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.

If you have configured TLS or SSL for your Oracle Beehive deployment, follow the steps described in ["Configuring Oracle Beehive Integration for Zimbra"](#) (these steps are similar to ["Configuring SSL with Self-Signed Certificates During Installation of Oracle Beehive"](#)).

Configuring Remote Voice Conferencing Media Server for Oracle Beehive Conferencing

Oracle Beehive comes with the Voice Conferencing Media Server, which provides the voice conferencing functionality for Oracle Beehive Conferencing. To improve the performance of the Oracle Beehive server and the general quality of voice conferences, Oracle recommends that you install the Voice Conferencing Media Server in its own dedicated computer and then configure your Oracle Beehive instance to use that Voice Conferencing Media Server remotely.

The Voice Conferencing Media Server is only available for Oracle Beehive for Linux x86. Consequently, you must configure Oracle Beehive for Solaris Operating System (SPARC 64-Bit) or for Microsoft Windows to use a remote Voice Conferencing Media Server from an Oracle Beehive for Linux x86 instance.

The following steps describe how to configure Oracle Beehive to use a Voice Conferencing Media Server remotely from another Oracle Beehive instance:

1. Install Oracle Beehive for Linux x86; use the same database as the one your current Oracle Beehive instance uses.
2. From your current Oracle Beehive instance, in the `_ConferenceService` component, set the `OwcUseRemoteMediaSessions` to `true`:

```
beectl modify_property --component _ConferenceService
--name OwcUseRemoteMediaSessions
--value true
```

3. Activate the configuration:

```
beectl activate_configuration
```

4. Restart the BEEAPP component in both your current Oracle Beehive instance and the instance that you just installed for its Voice Conferencing Media Server:

```
beectl status
```

Component identifier	Component type	Status
BTI_instance1.example.com	BTI	RUNNING
BEEAPP_instance1.example.com	OC4J	RUNNING
BEEEMGMT_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 BEEAPP_instance1.example.com
```

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 Beekeeper Post-Installation Procedures

If you have installed Oracle Beekeeper, perform one or more of the procedures described in "[Oracle Beekeeper Post-Installation Procedures](#)", depending on your security requirements or any other issue particular to your deployment.

Oracle Beehive Database Post-Installation Procedures

This chapter covers the following topic:

- [Modifying Database Connect String, Schema Name, or Credentials](#)

Modifying Database Connect String, Schema Name, or Credentials

Perform the following steps to modify the database connect string, schema name, or credentials of Oracle Beehive:

1. From your Oracle Beehive home, run the following command to specify the new connect string of the central configuration repository (which is also known as the bootstrap database):

```
beectl modify_bootstrap_configuration --connect_string <new connect string>
```

Note: The database connect string cannot contain any new line characters (`\n`, `\r`, or `\n\r`); the connect string must be a single-line value.

Run the following command to specify the new schema name:

```
beectl modify_bootstrap_configuration --schema_name <new schema name>
```

Run the following command to specify the new schema password:

```
beectl modify_bootstrap_configuration --schema_name <obfuscated new password> --obfuscated
```

To obfuscate a password, use the `beectl obfuscate` command:

```
beectl obfuscate
--expiration_time_in_minutes 0
Enter value for password:
```

```
Successfully obfuscated the string.
```

Alternatively, you may run `beectl` in shell mode and specify the password in plain text.

2. Activate the configuration:

```
beectl activate_configuration
```

3. Modify local configuration files:

```
beectl modify_local_configuration_files
```

4. From your Oracle Beekeeper home, run the Perl script `<Oracle Beekeeper home>/beehive/seed/clientconfig/sync_configuration.pl`. The following table describes the options for this script:**Table 16–1** Options for `sync_configuration.pl`

Option	Mandatory/ Optional	Description
<code>--oracle_home</code>	Mandatory	Oracle home path for Oracle Beekeeper instance
<code>--connect_string <new connect string></code>	Optional	New connection string. Note: The database connect string cannot contain any new line characters (<code>\n</code> , <code>\r</code> , or <code>\n\r</code>); the connect string must be a single-line value.
<code>--schema_name <new schema name></code>	Optional	New schema name
<code>--schema_password</code>	Optional	New schema password for the given schema name. Note: Do not specify the schema password with this option. The script will prompt you for the password.
<code>--help</code>	Optional	Outputs a help message describing this script's options
<code>--log_level <log level></code>	Optional	Sets the log level for this script. Valid values are ERROR, WARNING, INFO, CONFIG, FINE, and FINER.

For example, if you are changing the connect string, call the script as follows:

```
perl <Oracle Beekeeper home>/beehive/seed/clientconfig/sync_configuration.pl
--oracle_home <Oracle Beekeeper home full path name> --connect_string <new
connect string>
```

If you are changing the schema password, call the script as follows:

```
perl <Oracle Beekeeper home>/beehive/seed/clientconfig/sync_configuration.pl
--oracle_home <Oracle Beekeeper home full path name> --schema_name <new schema
name> --schema_password
```

The script will prompt you for the password.

Note: The Perl executable is located in `<Oracle Beekeeper home>/oracle/home/perl/bin`.

5. Perform steps 1 to 3 for all your non-Oracle Beekeeper and non-DMZ instances.

Configuring FTP

This chapter describes how to perform the following tasks:

- [Enabling Active Mode FTP](#)
- [Opening Ports Required by Oracle Beehive for FTP](#)
- [Configuring Passive Mode FTP](#)
- [Setting other FTP Parameters](#)

Enabling Active Mode FTP

By default, active mode FTP (FTP in PORT mode) connections to your Oracle Beehive instances are disabled. Follow these steps to enable them:

1. Set the `ActiveModeEnabled` property in the `_FtpService` property to `true`:

```
list_properties --component _FtpService
```

```
-----+-----
Property name      | Property value
-----+-----
ActiveModeEnabled  | false
-----+-----
Alias              | _FtpService
...

```

```
beectl modify_property
--component _FtpService
--name ActiveModeEnabled
--value true
```

2. Activate changes:

```
beectl activate_configuration
```

Note: If the `beectl activate_configuration` command asks you to run the `beectl modify_local_configuration_files` command, then run this command. It will then ask you to run the command on all your other instances. Run this command on all your instances.

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 17–1 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	<p>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.</p> <p>The property MaxDCPortCount indicates the number of ports in the range.</p> <p>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.</p>

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
Alias	_FtpService
ChannelSecurity	PLAIN_AND_TLS
DMSInstrumentation	false
DataConnectionPort	12121
DefaultCommandCharacterSet	UTF-8
DefaultDataPort	2120
FtpServerEnabled	true
InternalIP	DEFAULT
LightweightThreadCount	
ListeningPoint	DEFAULT
MaxDCPortCount	200
Port	2121
ServiceApplication	svcapp_ftp
ServiceInstances	instance_ftp_BEEAPP_site1.example.com
SessionTimeout	900000
Site	_CURRENT_SITE
Status	ENABLED
UploadBufferSize	260096

To change a property (such as the Port property), call the `beectl modify_property` command on all of your instances:

```
beectl modify_property
  --component _FtpService
  --name Port
  --value <new port value>
```

To save the changes and activate them, call the following command:

```
beectl activate_configuration
```

Note: If the `beectl activate_configuration` command asks you to run the `beectl modify_local_configuration_files` command, then run this command. It will then ask you to run the command on all your other instances. Run this command on all your instances.

Configuring Passive Mode FTP

You can configure the Oracle Beehive FTP service so that the IP address it sends as a PASV response depends on where the client initiated the passive mode FTP connection, either from within your network (your intranet) or outside your enterprise (the Internet).

This involves configuring two network interface groups: an internal one and an external one. A network interface group is the point of interconnection between BTI (Oracle Beehive Transport Interface) and a private or public network. A network interface group is mapped to a list of network interfaces.

Consequently, you may configure an internal network interface group and associate it with an IP address of your load balancer that is facing your intranet, and an external network interface group with another load balancer facing the Internet. If the FTP service accepts a client's passive mode FTP connection, the service will send the IP address of the appropriate load balancer, depending on where the client's FRP request originated (internally or externally).

Follow these steps to configure an internal and external network group and the FTP services so that it recognizes these groups:

1. Specify the internal and external network groups with the following commands. This command assumes that 140.87.24.44 is the IP address of your Internet-facing load balancer and 127.0.0.1 the IP address of your intranet-facing load balancer:

```
./beectl list_components --type BtiRedirector
```

```
-----+-----
Component type | Component identifier
-----+-----
BtiRedirector   | BTI_redirector_site1.example.com
-----+-----
```

```
beectl modify_property
--name AdditionalExecArgs
--value ESCAPE:-interface-groups
--value "EXTERNAL=140.87.26.44;INTERNAL=127.0.0.1"
--component BTI_redirectory_site1.example.com
```

2. Ensure that the BTI redirector is running with your new configuration with the `ps` command:

```
ps -Afw | grep redir
00:00:00 /my_oracle_home/beehive/bin/redirector -log-dir
/my_oracle_home/beehive/logs/bti -log-level all -xsl-dir
/my_oracle_home/beehive/xsl -ipc-path
/tmp/BTI-f871faff-90d3-4d5f-a94d-b13124072092 -server-port 21300 -bp-id-base
131072 -interface-groups EXTERNAL=140.87.26.44;INTERNAL=127.0.0.1
-bti-process 0 -bti-marker @@f871faff-90d3-4d5f-a94d-b13124072092
-btimon-port 3042 -btimon-token 1000000000000000
```

- Depending on your deployment, set one or more of the following parameters in the FTP service:

Table 17-2 FTP Parameters

Parameter	Description
DataconnectionPort	Starting port number from which the FTP server accepts data connections in passive mode.
MaxDcPortCount	Maximum number of data ports used for passive mode listening.
ListeningPoint	<p>Controls the response of the PASV command. Valid values for this parameter are INTERNAL, EXTERNAL, DEFAULT, and INTERNAL_AND_EXTERNAL.</p> <p>If ListeningPoint is set to DEFAULT, then the IP address of the application tier on which the FTP service is running is sent as the PASV response.</p> <p>If both ListeningPoint is set to INTERNAL and you configured a network interface group named "INTERNAL", then the IP address specified in the InternalIp address is sent as the PASV response. Otherwise, the IP address of the application tier on which the FTP service is running is sent.</p> <p>If both ListeningPoint is set to EXTERNAL and you configured a network interface group named "EXTERNAL", then the IP address specified in your virtual server is sent. Otherwise, the IP address of the application tier on which the FTP service is running is sent.</p> <p>Refer to "Installing Oracle Beehive in High Availability Environment" to configure the virtual server.</p> <p>If ListeningPoint is set to INTERNAL_AND_EXTERNAL, the FTP service will behave as if ListeningPoint were set to both INTERNAL and EXTERNAL.</p>
InternalIp	The IP address sent as the PASV response if ListeningPoint is set to INTERNAL (or INTERNAL_AND_EXTERNAL) and a network interface group named "INTERNAL" exists.

- Save the your changes and activate them, call the following command:

```
beectl activate_configuration
```

- Redeploy the FTP service with the following commands:

Note: If you change any of the FTP service parameters described in these steps (DataconnectionPort, MaxDcPortCount, ListeningPoint, or InternalIp) you must redeploy the FTP service (or restart the BEEAPP component) afterwards.

```
beectl list_components --type FtpServiceInstance
```

```
-----+-----
Component type | Component identifier
-----+-----
FtpServiceInstance | instance_ftp_BEEAPP_site1.example.com
-----+-----
```

```
1 Record(s) displayed.
```

```
beectl upgrade_service_instance
--service_instance_id instance_ftp_BEEAPP_site1.example.com
```

Setting other FTP Parameters

You must redeploy the FTP service (or restart the BEEAPP component) if you change one of the following FTP service parameters:

- DataconnectionPort
- MaxDcPortCount
- SessionTimeout: Amount of time between activity before the connection times out; default is 900 seconds or 15 minutes. Unit of measure is milliseconds.
- Port
- ListeningPoint
- InternalIp

Configuring Oracle Beehive Demilitarized Zone Instances

This chapter describes how to configure Oracle Beehive DMZ instances. It includes the following steps:

- [Step A: Configuring Oracle Wallet with Oracle Beehive DMZ Instances](#)
- [Step B: Configuring Oracle Beehive DMZ Instances](#)

If you want to manually uninstall a DMZ instance, have made an error while configuring your DMZ instances, or would like to completely rollback any DMZ configuration steps without uninstalling your DMZ instances, refer to "[Manually Deleting DMZ Instances](#)".

Note: If you want to configure your DMZ instances with SSL, follow the directions described in "[Configuring SSL with Oracle Beehive DMZ Instances](#)" 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. Enable auto login mode for the default wallet with the following command. The default password for the default wallet is welcome:

```
<Oracle home of DMZ instance>/bin/orapki wallet create  
-wallet <Oracle home of DMZ instance>/Apache/Apache/conf/ssl.wlt/default  
-auto_login  
-pwd welcome
```

Note: Alternatively, you may create a new wallet with auto login mode enabled. Use the same command except specify a different directory that does not contain a wallet. You may specify any password when creating a new wallet.

3. Edit the file `<Oracle home of DMZ instance>/beehive/conf/bti.properties` and change the value of `WalletDir` to the wallet directory you configured 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 configured in Step 2:

```
<notification-server interface="ipv4">  
    <!-- ... -->  
    <ssl enabled="true"  
        wallet-file="<Oracle home of DMZ instance>  
            /Apache/Apache/conf/ssl.wlt/default"/>
```

5. Restart the DMZ instance:

```
opmnctl stopall  
opmnctl startall
```

Step B: Configuring Oracle Beehive DMZ Instances

1. 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"](#).
2. If you have configured your Oracle Beehive DMZ instances for SSL as described in ["Configuring SSL with Oracle Beehive DMZ Instances"](#) in ["Configuring SSL"](#), then ensure that SSL is enabled for Oracle Notification Services (ONS) for your non-DMZ Oracle Beehive instances. Ensure that the property `NotificationServerSslEnabled` is `true` in the component `_current_site:OpmnCluster`. To set this property, run the following commands on your non-DMZ instances:

```
beectl modify_property  
    --component _current_site:OpmnCluster  
    --name NotificationServerSslEnabled  
    --value true  
    --activate_configuration  
  
<Oracle Beehive home>/opmn/bin/opmnctl stopall  
<Oracle Beehive home>/opmn/bin/opmnctl startall
```

3. If you have **not** configured your Oracle Beehive DMZ instances for SSL and specifically do not want to your Oracle Beehive DMZ instances for SSL, then follow this step.

By default, Oracle Beehive DMZ instances are configured to receive secure ONS notifications. If you do not want to use SSL, you must configure ONS to receive non-SSL notifications.

To do this, 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"/>
```


4. 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* (of a non-DMZ application tier). The OPMN port is specified by the *remote* attribute of the *port* element in the *notification-server* element.

5. 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
```

6. 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*

7. 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).

Note: 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.

8. On any Oracle Beehive non-DMZ instance, **but not on a DMZ instance**, run the following commands:

- a. Add the first DMZ instance to the non-DMZ instance by calling the `beectl add_dmz_home_instance` command on a 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>
```

- b. After the successful completion of the `beectl add_dmz_home_instance` command, immediately run the command `beectl activate_configuration` on the same non-DMZ instance. Do not execute any other other `beectl` commands on any other non-DMZ instance.
 - c. Repeat steps a and b for each of your other DMZ instances.
9. Run the following command on all the other non-DMZ instances:
- ```
beectl modify_local_configuration_files --log_level FINEST
```

## Configuring Oracle Application Server Single Sign-On with Oracle Beehive DMZ Instances

If you configured Oracle Beehive with Oracle Application Server Single Sign-On (OSSO) as described in ["Configuring Oracle Application Server Single Sign-On with Oracle Beehive"](#), then follow these steps for each of your Oracle Beehive DMZ instances to configure them with OSSO. For more information, refer to Chapter 4, "Configuring and Administering Partner Applications" in *Oracle Application Server Single Sign-On Administrator's Guide*.

1. Copy the configuration file that you created when you ran the OSSO registration tool (`<OSSO home>/sso/bin/ssoreg.sh`) to a directory on the computer hosting your Oracle Beehive DMZ instance. These steps assume that you copied this file as `/scratch/osso/osso.example.conf`.
2. Edit the file `<Oracle Beehive DMZ home>/Apache/Apache/conf/mod_osso.conf` and add the following line, which specifies the location of the OSSO configuration file you copied in the previous step:

```
OssosConfigFile /scratch/osso/osso.example.conf
```

3. Edit the file `<Oracle Beehive DMZ home>/Apache/Apache/conf/mod_osso.conf` and uncomment the following line:

```
include "<Oracle Beehive DMZ home>/Apache/Apache/conf/mod_osso.conf"
```

4. Restart OPMN on your Oracle Beehive DMZ instance:

```
<Oracle Beehive DMZ home>/opmn/bin/opmnctl stopall
<Oracle Beehive DMZ home>/opmn/bin/opmnctl startall
```

## 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.

## Manually Deleting DMZ Instances

Follow these steps to manually delete a DMZ instance.

---

**Note:** If you have made an error while configuring your DMZ instances or you would like to completely rollback any DMZ configuration steps without uninstalling your DMZ instances, then perform steps 3 and 4.

If you use the Install Wizard to uninstall DMZ instances, steps 3 and 4 are unnecessary.

---

1. From the computer hosting the DMZ instance, uninstall it with the Install Wizard:

```
runInstaller -uninstall
```

2. From any Oracle Beehive home, run the following command for each DMZ instance you want to delete:

```
beectl delete_dmz_home_instance --id <DMZ instance identifier>
```

To retrieve a list of DMZ instance identifiers, run the command `beectl list_components --type UnmanagedBeehiveInstance --all_visibilities`.

3. After the successful completion of the `beectl delete_dmz_home_instance` command, immediately run the command `beectl activate_configuration` on the same non-DMZ instance. Do not execute any other other `beectl` commands on any other non-DMZ instance.

4. For each DMZ instance you want to delete, edit the file *<Oracle home of DMZ instance>/opmn/conf/opmn.xml* and delete the topology information. The topology information is contained in the `<topology>` element. Delete this element. It will look similar to the following:

```
<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>
```

5. Restart each DMZ instance whose `opmn.xml` file you edited with the following commands:

```
<Oracle home of DMZ instance>/opmn/bin/opmnctl stopall
<Oracle home of DMZ instance>/opmn/bin/opmnctl startall
```

6. From each of your Oracle Beehive homes, run the following command:

```
beectl modify_local_configuration_files
```

---

## Configuring SSL

This chapter 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](#)
- [Installing Non-SSL Oracle Beehive Site](#)
- [Installing Oracle Internet Directory in SSL mode](#)

---

**Note:** Refer to "[Configuring Oracle Beekeeper for SSL Access](#)" to configure SSL for Oracle Beekeeper.

If you do not want to use SSL with your Oracle Beehive deployment, follow the steps described in "[Installing Non-SSL Oracle Beehive Site](#)".

If you have a load balancer supports SSL termination or offloading, you may offload SSL processing to your load balancer so that your Oracle Beehive instances do not have to decrypt SSL-encrypted data, thereby reducing the load of your Oracle Beehive instances. Refer to "[Configuring SSL Termination at Load Balancer](#)" in "[Installing Oracle Beehive in High Availability Environment](#)" for more information.

---

### 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 Self-Signed Certificates During Installation of Oracle Beehive](#)
- [Configuring SSL with Self-Signed 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. Install your first Oracle Beehive instance, if you have not already done so.
2. By default, an Oracle wallet with test certificates for OPMN is created in Oracle Beehive. This Oracle wallet is located in the following location:  
  
`<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.
3. Configure TLS on your first Oracle Beehive instance. Refer to "[Configuring TLS with Oracle Wallet](#)".
4. 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](#)".
5. 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](#)".
6. 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](#)".
7. Enable ORMIS on all your Oracle Beehive instances. Refer to "[Enabling ORMIS with Password-Protected Oracle Wallet](#)" in "[Configuring TLS with Oracle Wallet](#)".
8. Enable AJP/S on all your Oracle Beehive instances. Refer to "[Enabling AJP/S](#)".

---

**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 365 > 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 Self-Signed Certificates During Installation of Oracle Beehive

The following steps describe how to configure SSL with self-signed certificates during the installation of one or more Oracle Beehive instances:

1. 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.
2. 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"](#).
3. Install your first Oracle Beehive instance.
4. Configure TLS on your first Oracle Beehive instance. Refer to ["Configuring TLS with Oracle Wallet"](#).
5. 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`.
6. 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 the following location:

```
<Oracle Beehive home>/Apache/Apache/conf/ssl.wlt/default.
```

7. Perform the post-install steps for configuring Oracle RAC except Step 7 (Register for ONS Notification).
8. 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"](#).

9. Install an additional Oracle Beehive instance (software only install). In the following steps, this instance will be referred to as the second instance.
10. 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"](#).
11. 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.
12. Repeat Step 8 for the second instance.
13. Run the Config Wizard for the second instance and complete the configuration.
14. Configure TLS on all Oracle Beehive instances.
15. If you want to install another Oracle Beehive instance, repeat Steps 11 to 15.
16. Enable ORMIS on all Oracle Beehive instances. Refer to ["Enabling ORMIS with Password-Protected Oracle Wallet"](#) in ["Configuring TLS with Oracle Wallet"](#).
17. Enable AJP/S on all Oracle Beehive instances. Refer to ["Enabling AJP/S"](#).

## Configuring SSL with Self-Signed Certificates After Installation of Oracle Beehive

The following steps describe how to configure SSL with self-signed certificates after the installation of one or more Oracle Beehive instances:

1. 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.
2. 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"](#).
3. 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"](#).
4. 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`.
5. 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 the following location:

`<Oracle Beehive home>/Apache/Apache/conf/ssl.wlt/default`.



6. If you have multiple Oracle Beehive instances, repeat Steps 4 to 7 for each of your instances.
7. Enable ORMIS on all Oracle Beehive instances. Refer to ["Enabling ORMIS with Password-Protected Oracle Wallet"](#) in ["Configuring TLS with Oracle Wallet"](#)
8. Enable AJP/S on all Oracle Beehive instances. Refer to ["Enabling AJP/S"](#).

## 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 Self-Signed 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 Self-Signed Certificates After Installation of DMZ Instances

The following steps describe how to configure SSL with self-signed 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:

- [Creating Self-Signed Certificate and Importing it into Wallet](#)
- [Creating CA-Signed Certificate and Importing it into Wallet](#)

### 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 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 365
```

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 365 > server.crt
```

This command generates two files, `cacert.srl` 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 365
```

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 365
```

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 365
```

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 -cert user_certificate.txt
```

## Installing Non-SSL Oracle Beehive Site

The following steps describe how to install a non-SSL Oracle Beehive site in which none of its tiers communicate using SSL:

---

---

**Note:** Because Oracle Beehive DMZ instances have SSL enabled by default, the following steps will not work for DMZ instances unless you configure them to receive non-SSL notifications as described in ["Step B: Configuring Oracle Beehive DMZ Instances"](#) in ["Configuring OracleBeehiveDemilitarizedZoneInstances"](#).

---

---

1. Install your first Oracle Beehive application tier. Note that this application tier, by default, will have SSL disabled for Oracle Notification Service (ONS), which is used by OPMN of this application tier to communicate with other OPMNs in the site. In the next step, you will disable SSL (if necessary).
2. Ensure that the value of `NotificationServerSslEnabled` in the `_current_site:OpmnCluster` component in the first Oracle Beehive application tier is false:

```
beectl list_properties
--component _current_site:OpmnCluster
--name NotificationServerSslEnabled
```

If NotificationServerSslEnabled is true, then set it to false:

```
beectl modify_property
--component _current_site:OpmnCluster
--name NotificationServerSslEnabled
--value false
--activate_configuration
```

3. In the first Oracle Beehive application tier, set the value of HttpServerSslEnabled in the \_current\_site:HttpServerCluster component to false, then run beectl modify\_local\_configuration\_files:

```
beectl modify_property
--component _current_site:HttpServerCluster
--name HttpServerSslEnabled
--value false
--activate_configuration
```

```
beectl modify_local_configuration_files
```

4. Install any additional Oracle Beehive application tiers. You do not need to perform any additional steps for these application tiers.

## Installing Oracle Internet Directory in SSL mode

Oracle Identity Management Infrastructure and Oracle Identity Federation are supported on the following Operating System versions:

- Red Hat Enterprise Linux AS Release 4
- Red Hat 2.1
- Red Hat 3
- SuSE 9
- UnitedLinux 1.0

The following steps describe how to install Oracle Internet Directory in a Secure Sockets Layer (SSL):

1. Download Oracle Identity Management Infrastructure and Oracle Identity Federation from the following URL:

<http://www.oracle.com/technetwork/middleware/ias/downloads/101401-099957.html>

- Download x86 version from Linux column (Both Disk1 and Disk2) to an empty directory. (Prefer a subdirectory of /scratch/\$USER/)
- To verify the integrity of the downloaded file, after the file has been transferred to a Unix host, run cksum and compare with the cksum information listed on the download page.
- Go to the directory and extract the contents by using the following command.

```
cpio -idvm < <Disk1 cpio_file>
cpio -idvm < <Disk2 cpio file>
```

2. If an old installation exists, reboot the system and remove the directory containing the previous OID installation.
3. Start the installation of OID using Disk1/runInstaller. Execute oraInstRoot.sh present in oraInventory directory to start the installation.

Click Next wherever no input is asked.

- On the **Specify File Locations** screen, change the Destination Path to your Oracle Home.
- On the **Select a Product to Install** screen, select **Oracle Application Server Infrastructure 10g** and click Next.
- On the **Select Installation Type** screen, select **Identity Management and Metadata Repository**.
- On the **Product-specific Prerequisite Checks** screen, let the check complete. It may show one warning. Select the checkbox **Checking security kernel parameters** and click Next.
- On the **Confirm Pre-Installation Requirement** screen, select the checkbox **Root Privileges** and click Next.
- On the **Select Configuration Options** screen, select the checkbox **Oracle Application Server Certificate Authority (OCA)** and click Next.
- On the **Specify Port Configuration Options** screen, select **Automatic** and click Next.
- On the **Specify Namespace in Internet Directory** screen, leave the **Suggested Namespace** selected and click Next.
- On the **Specify OCA Distinguished Name** screen, enter "test" in all three textboxes under **Typical DN**, leave it selected and click Next.
- On the **Specify OCA Key Length** screen, select the needed **Key Length** and click Next. Prefer '1024' unless otherwise needed.
- On the **Specify OCA Administrator's Password** screen, specify the password as "Welcome1" and click Next.
- On the **Specify Database Configuration Options** screen, ensure that the database file location is a subdirectory of \$ORACLE\_HOME and click Next. (You may want to note down the details on this page.)

---

**Note:** If the specified SID already exists choose a name of your choice.

---

- On the **Specify Database Schema Passwords** screen, select **Use the same password for all the accounts**, specify the password as "Welcome1" and click Next.
  - On the **Specify Instance Name and ias\_admin Password** screen, specify a suitable instance name, specify the password as "Welcome1", confirm it and click Next.
4. The Confirmation screen displays the summary of what you have selected, check the details and click install.
  5. When the installation starts, if your disk 2 is not in the same directory, the system will prompt you for the location of the disk 2 directory, specify the path and proceed further.
  6. In between the installation, the system will ask you to run a script as a root user, run the script and click OK on the pop-up message.
  7. Save the configuration details in a file for reference.

**To test the installation:**

- Run the following script to check whether ldapbind works:

```
$ORACLE_HOME/bin/ldapbind -D cn=orcladmin -w Welcome1 -U 1 -h <OID_hostname> -p
<port>
```

- "-U 1" is for non-SSL mode
- See Appendix to find out the port

**Configuring the OID in SSL mode**

1. Create an Auto Login Wallet with self signed user certificate by running the following commands:

```
setenv ORACLE_HOME /scratch/$USER/OraHome_1
cd $ORACLE_HOME/bin
orapki wallet create -wallet /home/$USER/ORACLE/WALLET -auto_login -pwd
Welcome1
orapki wallet add -wallet /home/$USER/ORACLE/WALLET -keysize 1024 -dn "cn=<OID_
hostname>" -self_signed -validity 365
```

2. Add a new configuration set:

- a. In the navigator pane, select **Oracle Internet Directory Servers**, then **Directory Server** instance, and then select **Server Management**.
- b. Select **Directory Server**. The numbered configuration sets are listed beneath your selection.
- c. Right click on the **Configuration Set 1** and select **Create Like**.
- d. In the new configuration set window, enter Non SSL port that is not already in use.
- e. Select the **SSL Settings** tab, modify the fields as described below:  
 SSL Authentication: SSL Server Authentication  
 SSL Enable: Both SSL and Non-SSL  
 SSL Wallet URL: file://home/<username>/ORACLE/WALLET  
 SSL Port: 1636 (Any unused port)
- f. Click Ok.

You can review the settings by clicking the newly created configuration set node.

- g. Exit the Oracle Directory Manager.

3. Start a new instance by running the following command:

```
$ORACLE_HOME/bin/oidctl connect=orcl server=oidldapd instance=2 configset=2
start
```

4. Test if the SSL is working by running the following command:

```
$ORACLE_HOME/bin/ldapbind -p 1636 -U 2 -W file:/home/$USER/ORACLE/WALLET -P
Welcome1 -h <hostname>
```

---

---

**Note:** If the wallet does not contain any user certificate or if there is a mismatch in certificate.

```
$ldapbind -p 1636 -U 2 -W file:/home/$USER/ORACLE/WALLET -P
welcome1
Unknown Error Encountered
```

If you make any changes in the configuration set, you must restart the instance that is running the configuration set for the changes to take effect by running the following commands:

```
oidctl connect=orcl server=oidldapd instance=nn configset=<config
no.> stop
oidctl connect=orcl server=oidldapd instance=nn configset=<config
no.> start
```

---

---

## Appendix

### 1. Shell variables that need to be set:

- \$ORACLE\_HOME=/scratch/\$USER/OraHome\_1
- \$ORACLE\_SID=orcl

### 2. Find out the ports on which the OID is listening:

#### a. Run the following command:

```
ps -ef | grep oidldapd
```

#### b. Look for options -port and -sport.

### 3. Start the OID manager:

#### a. Run the following command to start the OID manager:

```
$ORACLE_HOME/bin/oidadmin
```

#### b. Enter the following details:

- User: orcladmin
- Password: Welcome1
- Add a new server with hostname and port by clicking the icon against Server.

---

---

**Note:** Port number can be found out by running the following command:

```
ps -ef | grep oidldapd
```

---

---

- Click Login.

### 4. To restart OID after rebooting the system:

#### a. Place the following content in a file /scratch/\$USER/.ENV

```
setenv ORACLE_HOME /scratch/$USER/OraHome_1
setenv ORACLE_SID orcl
setenv PATH ${PATH}:${ORACLE_HOME}/bin
```

#### b. Use the following command to start the database:

```
source /scratch/$USER/.ENV
```



```
sqlplus "sys/Welcome1 as sysdba" <<EOF
?startup
?EOF
lsnrctl start
```

- c. Use the following command to all instances of OID:

```
oidmon connect=orcl start
```

- d. Use the following command to start the only one of the instances of OID:

```
oidctl connect=orcl server=oidldapd instance=nn configset=cf start
```

## Configuring Beehive to connect to OID in non-SSL and SSL modes

Before you begin to configure Beehive to connect to OID, ensure the following:

- **For non-SSL mode:**

- OID is installed in the \$ORACLE\_HOME folder.
- ldapbind is working correctly; to check, the following command should give output as "Bind Successful".

```
$ORACLE_HOME/bin/ldapbind -D cn=orcladmin -w Welcome1 -U 1 -h <OID_
hostname> -p <port>
```

- \$MWH is Beehive installation Directory.

- **For SSL mode:**

- Include \$ORACLE\_HOME/bin in \$PATH environment variable.
- OID is listening only on SSL-port in SSL-only mode.
- /home/\$USER/ORACLE/WALLET contains self signed certificate of OID with cn = hostname of OID, and the same certificate in the trusted certificate list.

- ldapbind is working correctly; to check, following command should give output as "Bind Successful"

```
ldapbind -p <SSL-port> -U 2 -W file:/home/$USER/ORACLE/WALLET -P Welcome1
```

- \$MWH is Beehive installation Directory.
- Keytool utility is available, if not, include the bin directory of jdk6 or later in the \$PATH environment variable.
- OID entries and profile entries match. See Initial configuration in non-SSL mode.

### To configure Beehive to connect to OID (in non-SSL mode):

1. Set the initial configuration. Refer the [Appendix](#) section to start oiddas. Skip this step if already done.

If group Extpts exists in your ldap\_profile.xml, add the group to OID using oiddas.

2. Make the following necessary changes in ldap\_profile.xml file; skip the step if already done.
  - Change the host name, port number, and SSL-port to match the OID installation. Refer the [Appendix](#) section to find out the port numbers.

- Add a tag `<user_objectclass>person</user_objectclass>` in the xml file inside `<ldap_server>` tag after `<group_search_base>` tag. If the tag exists, skip the step.
- Comment out the entry `<directory_attribute_map_entry>` which has the tag `<source_object>EXTENDED_ENTERPRISE_USER</source_object>`.

- Run the following command:

```
beectl>list_organizations --scope enpr=Oracle
Organization name: <orgn_name>
```

In the xml file, search for "orgn=" and modify the Organization name by the name listed in the above command.

- Run the following commands:

```
$cd $MWH/instance_Oracle_BH1/beehive/bin
$./beectl
beectl> obfuscate --expiration_time_in_minutes 0
```

Enter the value of password as Welcome1.

Use the string generated to replace the value of `<ldap_user_password>` tag in xml file.

3. Make the Beehive installation (both 32-bit and 64-bit) to use the OID 32-bit for authentication:

```
beectl> add_directory_profile --file <PATH_TO_PROFILE>/oidprofile.xml
beectl> modify_property --component _AuthenticationService --name AuthStoreType
--value ldap
beectl> activate_configuration
beectl> modify_local_configuration_files
beectl> validate_directory_entry --all_users --profile ldap_profile
beectl> validate_directory_entry --all_users --profile ldap_profile --commit
beectl> modify_local_configuration_files
```

4. To test the configuration:

- Add a user in OID using `oiddas`, refer the [Appendix](#) section.

For example, user ID "abcd", password: "Welcome1".

- Try to login from the command line by using the following command:

```
beectl> login --authuser junk --authpassword Welcome1
```

- Try to login from a browser application such as `/zimbra` or `/bconf`

1. Browse for <http://hostname:7777/zimbra>

2. On the login page provide username: "abcd" and password "Welcome1"

3. Click Login.

#### **To configure Beehive to connect to OID (in SSL mode):**

1. Save the trusted certificate of OID to a keystore.
  - a. Using Oracle Wallet Manager, open `/home/$USER/ORACLE/WALLET` and export trusted certificate with `cn=<OID_hostname>` to file `/some_directory/<OID_hostname>.cert`.

- b. Copy the exported certificate and `cwallet.sso` certificate to home directory of user on which Beehive is installed; `cwallet.sso` certificate is required to sync Beehive with OID.
- c. Start Oracle Wallet Manager (OWM) by executing `./owm` OWM can be located in `$ORACLE_HOME/bin`.
- d. Use the following command to add the exported certificate to a keystore.
 

```
keytool -importcert -trustcacerts -file <directory where the certificates
are copied on to beehive> -keystore ~/<OID_hostname>.jks
```
- e. Give password for the keystore as `Welcome1`.

2. Do step 1 and 2 of non-SSL mode if that is not done.

3. Add Oracle Wallet to Beehive by running the following command:

```
beectl>list_components --type BeehiveInstance
beectl> modify_property --component beehive_instance_
beehive.adc2171171.us.oracle.com --name WalletDir --value <Wallet Directory>
beectl> modify_secure_property --component beehive_instance_
beehive.adc2171171.us.oracle.com --name WalletPassword --value
<wallet-password> --activate_configuration
```

4. Add Keystore to the Beehive instance:

```
beectl> modify_property --component beehive_instance_
beehive.adc2110271.us.oracle.com --name KeystoreFile --value
/home/rnataraj/<OID_hostname>.jks
beectl> modify_secure_property --component beehive_instance_
beehive.adc2110271.us.oracle.com --name KeystoreFilePassword --value <key
store password> --activate_configuration
modify_local_configuration_files
```

5. Make the Beehive installation (both 32-bit and 64-bit) to use the OID 32-bit for authentication.

Modify the non-SSL port in the profile to unused port, run `modify_local_configuration_files` if `validate_directory_entry` fails.

```
beectl> add_directory_profile --file PATH_TO_PROFILE/oidprofile.xml
beectl> modify_property --component _CURRENT_SITE:LdapServer --name SslEnabled
--value true
beectl> modify_property --component _AuthenticationService --name AuthStoreType
--value ldap
beectl> activate_configuration
beectl> modify_local_configuration_files

beectl> validate_directory_entry --all_users --profile ldap_profile
beectl> validate_directory_entry --all_users --profile ldap_profile --commit
beectl> modify_local_configuration_files
```

6. If the profile was already added to Beehive in non-SSL mode, do the following steps:

a. To ensure that Beehive is connected to OID in SSL mode, change the non-SSL port of Ldap Server to unused port by using the following command:

```
beectl> modify_property --component _CURRENT_SITE:LdapServer --name
LdapServerPort --value <unused port>
```

b. Make sure the SSL port matches the instance of OID running in SSL port:

```
beectl> modify_property --component _CURRENT_SITE:LdapServer --name
LdapServerSslPort --value <SSL port>
```

- c. Enable the SSL mode by running the following command:

```
beectl> modify_property --component _CURRENT_SITE:LdapServer --name
SslEnabled --value true --activate_configuration
```

7. Run following commands:

```
beectl> validate_directory_entry --all_users --profile ldap_profile
beectl> validate_directory_entry --all_users --profile ldap_profile --commit
beectl> modify_local_configuration_files
```

8. Login to /zimbra with an OID user.

[http://beehive\\_hostname:7777/zimbra](http://beehive_hostname:7777/zimbra)

## Appendix

1. Shell variables that need to be set:

- \$ORACLE\_HOME=/scratch/\$USER/OraHome\_1
- \$ORACLE\_SID=orcl

2. Add a user to OID:

- a. Make sure OID is running on default configuration set. If not, run the following command.

```
$ORACLE_HOME/bin/oidctl connect=orcl server=oidldapd instance=1 configset=0
start
```

- b. Run \$ORACLE\_HOME/opmn/bin/opmnctl status to ensure that atleast OC4J\_SECURITY and HTTP\_Server are running. If not the start them using the following command:

```
$ORACLE_HOME/opmn/bin/opmnctl startproc process-type=OC4J_SECURITY
$ORACLE_HOME/opmn/bin/opmnctl startproc process-type=HTTP_Server
```

- c. Browse the following link:

<http://hostname:7777/oiddas/>

- d. Select **Directory** from the tabs, or click **Directory** link on the right side of the page.

- e. On the Login page, login using **orcladmin/Welcome1**.

- f. Click **Create** and fill the mandatory details to add the user.

- g. Click **Submit** to create the user.

3. Add a group to OID:

- a. After login to oiddas, click **Groups** in the horizontal panel.

- b. Click **Create** and enter the mandatory details to add the group.

- c. Click **Submit** to create the group.

4. To find out on which port the OID is listening, run the following command:

```
ps -ef | grep oidldapd
```

Look for the number after port for non-SSL and sport for SSL.

**5. Start the OID manager:****a. Run the following command to start the OID manager:**

```
$ORACLE_HOME/bin/oidadmin
```

**b. Fill up following details**

User: orcladmin

Password: Welcome1

Add a new server with a hostname and port by clicking the icon against Server.

---

---

**Note:** Port can be found out by running the following command:

```
ps -ef | grep oidldapd
```

---

---

**c. Click Login.****6. To restart OID after rebooting the system:****a. Place the following content in a file /scratch/\$USER/.ENV**

```
setenv ORACLE_HOME /scratch/$USER/OraHome_1
setenv ORACLE_SID orcl
setenv PATH ${PATH}:${ORACLE_HOME}/bin
```

**b. Use the following command to start the database:**

```
source /scratch/$USER/.ENV
sqlplus "sys/Welcome1 as sysdba" <<EOF
?startup
?EOF
lsnrctl start
```

**c. Use the following command to start all instances of OID:**

```
oidmon connect=orcl start
```

**d. Use the following command to start any one of the instances of OID:**

```
oidctl connect=orcl server=oidldapd instance=nn configset=cf start
```



---

## Configuring TLS with Oracle Wallet

This chapter describes how to configure Transport Layer Security (TLS) 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 or FTPS. The following steps configure Oracle Beehive to use Oracle Wallet so that clients may access Oracle Beehive with a TLS connection.

- [Step 1: Enabling Auto Login Mode for Default Wallet](#)
- [Step 2: Configuring Oracle Beehive Instance to Use Oracle Wallet](#)
- [Step 3: Replacing Test Certificates in 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: Enabling Auto Login Mode for Default Wallet

1. Ensure that the environment variable ORACLE\_HOME is set to the home directory of Oracle Beehive.
2. Enable auto login mode for the default wallet with the following command. The default password for the default wallet is welcome:

```
<Oracle home>/bin/orapki wallet create
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-auto_login -pwd welcome
```

---

**Note:** Alternatively, you may create a new wallet with auto login mode enabled. Use the same command except specify a different directory that does not contain a wallet. You may specify any password when creating a new wallet.

---

## 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`.

*<Oracle home>/Apache/Apache/conf/ssl.wlt/default* is the location of the auto login wallet you configured or created previously.

2. Activate the configuration and restart by running 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 `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.

---

---

## Step 3: Replacing Test Certificates in Oracle Wallet

If you created a wallet as part of TLS configuration, it will contain test certificates. These certificates are valid for a very short period of time and will expire quickly. Once they expire, when a user tries to access HTTPS, that user will receive an error similar to one of the following:

- You have received an invalid certificate.
- The security certificate presented by this Website has expired or is not yet valid.
- The connection is untrusted.

Consequently, you must replace these test certificates with self-signed or CA-signed certificates. Refer to the sections ["Creating Self-Signed Certificate and Importing it into Wallet"](#) and ["Creating CA-Signed Certificate and Importing it into Wallet"](#) in ["Configuring SSL"](#) for more information.

---

---

**Note:** After replacing these certificates with self-signed or CA-signed certificates, restart Oracle Beehive.

---

---

## Changing Oracle Wallet Password

Follow these steps to change the Oracle Wallet password:



1. Specify the wallet's new password in Oracle Beehive by running the following `beectl` commands:

```
beectl modify_property
--component beehive_instance_<instance>.<host name>
--name WalletDir
--value <$ORACLE_HOME>/Apache/Apache/conf/ssl.wlt/default
```

```
beectl modify_local_configuration_files
```

`<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`.

`--value` is the directory location of the wallet.

To obfuscate a password, use the `beectl obfuscate` command:

```
beectl obfuscate
--expiration_time_in_minutes 0
Enter value for password:
```

Successfully obfuscated the string.

2. Set the wallet password:

```
beectl modify_secure_property
--component beehive_instance_<instance>.<host name>
--name WalletPassword --value <password>
```

3. Change the password to the one you specified in the previous step in Oracle Wallet Manager, `<Oracle home>/bin/owm`. Refer to "Changing the Password" in Chapter 11, "Managing Wallets and Certificates" in *Oracle Application Server Administrator's Guide* for more information.

4. Activate the configuration and commit changes:

```
beectl activate_configuration
beectl modify_local_configuration_files
```

## 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

Oracle Remote Method Invocation over Secure Socket Layer (ORMIS) is ORMI over SSL. 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 --value <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 chapter 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 "Configure Oracle Internet Directory for SSL" in Chapter 13, "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 an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide*.

## 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
<b>LdapServer</b>	<b>oidldapdirectoryprofile_example.com</b>
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
<b>BeehiveInstance</b>	<b>beehive_instance_example.com</b>

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."

Follow these steps to eliminate this request:

1. Run the following beectl commands:

```
beectl list_components --type HttpServer
```

```
-----+-----
Component type | Component identifier
-----+-----
HttpServer | ohs_site1.example.com
-----+-----
```

```
beectl modify_property
--component ohs_site.example.com
--name DocRootLimitExcept
--value GET POST OPTIONS
```

2. Run the following commands to activate your changes:

```
beectl activate_configuration
beectl modify_local_configuration_files
```

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.





---

## 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.

This chapter covers the following topics:

- [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](#)

---

**Note:** 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	
<b>AssumeSingleEndpoint</b>	<b>true</b>

```
...
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:*:25 |
...
```

```
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
```

---

**Note:** 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
```

---

**Note:** 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)**.

4. Click **OK**.

## Configuring XMPP

Extensible Messaging and Presence Protocol (XMPP) is an open XML technology for presence and real-time communication. For users to authenticate against Oracle Beehive's XMPP Service, you must perform the following steps to configure it:

1. Follow the steps described in "[Configuring TLS with Oracle Wallet](#)", XMPP Service requires a TLS connection.
2. Set the DomainNames property of XMPP Service. By default, the value of DomainNames is `example.com`. Change it to the name of your domain. Call the following `beectl` commands.

These commands assume the name of your domain is `mydomain.com` and the component identifier of BEEAPP is `BEEAPP_mysite.mydomain.com`:

```
beectl list_properties --component _XmppService --name DomainNames
```

```
-----+-----
Property name | Property value
-----+-----
DomainName | example.com
-----+-----
```

```
beectl modify_property
--component _XmppService
--name DomainNames
--value mydomain.com
```

Changes to configuration repository are not activated.  
Successfully stored the property for component id  
a471ba52-b384-4b31-afe2-45ea8c38a658.

```
beectl activate_configuration
```

Proposed configuration is saved successfully and activated now.





---

## Configuring Oracle Application Server Single Sign-On with Oracle Beehive

---

This chapter describes how to register Oracle Beehive as a partner application with Oracle Application Server Single Sign-On (OSSO), which means that you may delegate the authentication function to the single sign-on server. For more information, refer to Chapter 4, "Configuring and Administering Partner Applications" in *Oracle Application Server Single Sign-On Administrator's Guide*.

---

**Note:** If you configured OSSO with HTTP, then disable SSL by running the following commands:

```
beectl list_components --type HttpServerCluster
beectl modify_property
 --component <identifier returned from the previous command>
 --name HttpServerSslEnabled --value false
beectl modify_property
 --component _VIRTUAL_SERVER
 --name HttpSslEnabled
 --value false
beectl activate_configuration
beectl modify_local_configuration_files
```

For more information about disabling SSL, refer to steps 2 and 3 in ["Installing Non-SSL Oracle Beehive Site"](#) in ["Configuring SSL"](#).

---

1. OSSO requires Oracle Internet Directory. Consequently, integrate Oracle Internet Directory with Oracle Beehive as described in "Integrating an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide*.
2. 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
```

3. Set the environment `ORACLE_HOME` to the full path of the directory of the Oracle product that comes with OSSO.
4. 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 OSSO (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 effective URL of your Oracle Beehive instance. Use the following format:

```
http[s]://<Oracle Beehive HTTP host>.<domain>:<port>
```

For example:

```
https://application.mydomain.com:4443
```

Omit the port number if the HTTP server is listening on the default HTTP port of 80 or the default HTTPS port of 4443. To determine the HTTP or HTTPS 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.
5. Copy the configuration file you created in the previous step (`osso.example.com`) to `<Oracle Beehive home>/Apache/Apache/conf/osso`. Rename the file to `osso.conf`.
  6. In the `_AuthenticationService` component, set the property `OssoConfigFile` to `<Oracle Beehive home>/Apache/Apache/conf/osso/osso.conf`, activate the configuration, then commit changes:

```
beectl modify_property
--component _AuthenticationService
--name OssoConfigFile
--value <Oracle Beehive home>/Apache/Apache/conf/osso/osso.conf
```

```
beectl activate_configuration
beectl modify_local_configuration_files
```

7. Restart the HTTP server:

```
beectl list_components --type HttpServer
```

```
-----+-----
Component type | Component identifier
-----+-----
HttpServer | ohs_site1.example.com
-----+-----
beectl restart --component ohs_site1.example.com
```

---

## Installing Oracle Beehive Extensions for Outlook

This chapter describes how to install and configure Oracle Beehive Extensions for Outlook.

Oracle recommends that you install and deploy Oracle Beehive Extensions for Outlook using Oracle Beehive's Device Management Service (DMS). This installation method is recommended for almost all desktops with standard environments and administrative privileges. A desktop based, non-DMS installation, deployment, and configuration method is available for custom environments and lockdown systems with limited administrative privileges. Locked down users may also install Oracle Beehive Extensions for Outlook using an Active Directory group policy.

This chapter covers the following topics:

- [System Requirements](#)
- [Device Management Service \(DMS\) Based Installation](#)
- [Desktop Based Installation \(Non-DMS Process\)](#)
- [Installing Oracle Beehive Extensions for Outlook Using Active Directory](#)
- [Using Oracle Beehive Extensions for Outlook as Profile Migration Tool](#)

### System Requirements

This section describes the software and hardware requirements for Oracle Beehive Extensions for Outlook. For updated list of certified hardware platforms and operating system version, review the certification matrix on the My Oracle Support Web site at the following URL:

<http://support.oracle.com/>

#### Operating System

Oracle Beehive Extensions for Outlook supports the following Microsoft® Windows® operating systems, running Microsoft Outlook 2003, 2007, 2010, and 2010 64-bit:

- Microsoft Windows XP - Home or Professional
- Microsoft Windows 7 (32-bit and 64-bit edition) - Home Premium, Professional, Ultimate or Enterprise
- Microsoft Windows 2003 Server
- Microsoft Windows 2008 Server
- Microsoft Windows 2008 Server R2

---

---

**Note:** You need administrative privileges to install Oracle Beehive Extensions for Outlook on Windows XP and Windows 2003 Server.

You must install Windows XP Service Pack 2 on Windows XP computers if you intend to use non-English locales with Oracle Beehive Extensions for Outlook. Attempting to run Oracle Beehive Extensions 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.

---

---

### Disk Space

- 200 MB minimum
- Hard disk usage varies according to configuration. Custom installation options may require more or less hard disk space. The amount of online data the user accesses may also affect the disk space requirement.

### RAM

Refer to the RAM requirements of your Microsoft Outlook client.

### Microsoft Outlook

- Microsoft Outlook 2010 (32-bit and 64-bit)
- Microsoft Outlook 2007 (with Service Pack 2)
- Microsoft Outlook 2003 (with Service Pack 3)

---

---

**Note:** Oracle certifies and recommends the latest service pack version for Microsoft Outlook. However, other service packs may also work with Oracle Beehive Extensions for Outlook.

You must install a native language version or Multilingual User Interface Pack (MUI) for non-English version localization.

---

---

## Device Management Service (DMS) Based Installation

Oracle recommends that you install and deploy Oracle Beehive Extensions for Outlook using Oracle Beehive's Device Management Service (DMS). This installation method is recommended for almost all desktops with standard environments and administrative privileges. It is designed for centralized installation, configuration, management of auto-updates, and deployment.

This section covers the following topics:

- [Installing Oracle Beehive Extensions for Outlook Using Remote Downloader](#)
- [Updating and Configuring Oracle Beehive Extensions for Outlook Through DMS](#)
- [Uninstalling and Downgrading Oracle Beehive Extensions for Outlook \(DMS Process\)](#)

### Installing Oracle Beehive Extensions for Outlook Using Remote Downloader

You may install Oracle Beehive Extensions for Outlook by distributing a remote downloader to end users. The remote downloader enables end users to download and install the provisioned Oracle Beehive Extensions for Outlook application from the

Device Management Service (DMS) repository. By default, the DMS contains a pre-seeded Oracle Beehive Extensions 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, refer to "Managing the Device Management Service" in *Oracle Beehive Administrator's Guide*. You may do this in silent mode. Refer to ["Installing Oracle Beehive Extensions for Outlook Using Remote Downloader Using Silent Mode"](#) for more information.

### Obtaining Remote Downloader

Obtain the Oracle Beehive Extensions for Outlook remote downloader, outlook\_extensions\_downloader.exe in the directory `<Oracle Beehive home>/beehive/bootstrap/obio/downloader`.

You may also obtain the remote downloader from Oracle Beehive Central, a Web-based client that provides users a central location to download supported clients and set their preferences for Oracle Beehive functionality.

### Pre-Seeding Server Name in 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 want 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

You may pre-seed the port number that Oracle Beehive Extensions for Outlook uses to connect to the server.

For example, if your remote downloader name is `faulkner.oracle.com.exe` and you want to pre-seed the port number in the downloader, rename the file to `faulkner.oracle.com,443.exe`, where 443 is your port number.

---

**Note:** The pre-seeded port number is the port number of the DMS on the server that is used to obtain the settings for direct and HTTPS connections.

---

## Installing Oracle Beehive Extensions for Outlook Using Remote Downloader Using Silent Mode

You may also use the Oracle Beehive Extensions for Outlook downloader to provide a full, non-interactive deployment option. To do so, use the silent switch, `/s`.

Specify the required connection parameters through the command line or in an initialization file.

**Specifying Connection Parameters Through Command Line** The following is an example of specifying connection parameters through the command line (line breaks have been added for clarity):

```
outlook_extensions_downloader.exe
 /server example.com
 /user OracleBeehiveUserName
 /direct port=5224 secure=true
```

```
/https port=443 secure=true
/s UI=1
```

The following table describes the Oracle Beehive Extensions for Outlook downloader's command line options:

**Table 25–1 Oracle Beehive Extensions for Outlook Downloader Options**

Option	Description
/server <host name>	Oracle Beehive server address
/user <user name>	Oracle Beehive user account
/pass <password>	User's password  <b>Note:</b> It is recommended that you use the option /s UI=1 instead of the /pass option. With the /s UI=1 option, the connection dialog is presented in which all the controls are disabled and only the password field is enabled.
/direct port=<port number> secure=<true/false>	Specifies the following for direct connection: <ul style="list-style-type: none"> <li>port: Valid port number to be used to establish a direct connection</li> <li>secure: Boolean value; if true, then use a secure connection</li> </ul>
/https port=<port number> secure=<true/false>	Specifies the following for HTTPS connection: <ul style="list-style-type: none"> <li>port: Valid port number to be used to establish an HTTPS connection</li> <li>secure: Boolean value; if true, then use a secure connection</li> </ul>
/create_profile	Available in Oracle Beehive Release 1 (1.5.1.2) or later. Determines if an Oracle Beehive Extensions for Outlook profile should be created on the user's computer during installation. This parameter may be assigned one of the following values: <ul style="list-style-type: none"> <li>0: Do not create a profile</li> <li>1: Always create a profile (either for a new install or an upgrade)</li> <li>2: Only create a profile for a new install (default value)</li> <li>3: Only create a profile if there is no profile with the exact name that exists on the user's computer (either for a new install or upgrade)</li> </ul>
/discover port=<port number>	Specifies the port number of the DMS on the server to obtain the settings for direct and HTTPS connections.  By using this option, you do not need to specify either the /direct or /https options.
/profile_name	Available in Oracle Beehive Release 1 (1.5.1.2) or later. Changes the name of the profile created during installation of Oracle Beehive Extensions for Explorer. The default profile name is Beehive.

**Table 25–1 (Cont.) Oracle Beehive Extensions for Outlook Downloader Options**

Option	Description
/repair	Available in Oracle Beehive Release 1 (1.5.1.2) or later.  Downloads and reinstalls the MSI package from the DMS if the installed version on the user's computer is the same version as the MSI package.  If /create_profile option has a value of 1 or 3, then the /repair option is used even if it is not specified.
/s	Use silent mode

**Specifying Connection Parameters in Initialization File** Specify the full path of the initialization file you want to use as follows. In this example, C:\testSilentDownloader.ini is the full path of the initialization file:

```
outlook_extensions_downloader.exe /s C:\testSilentDownloader.ini
```

If you use the silent switch with a UI=1 parameter, the connection dialog will be presented in which all the controls are disabled and only the password field is enabled:

```
outlook_extensions_downloader.exe /s UI=1 C:\testSilentDownloader.ini
```

The downloader will return a value of 1 or 0, which indicates whether the operations performed by it failed (1) or were successful (0). You may use this returned value in a script to determine whether or not the remote downloader was successful.

Use the following template to create a silent initialization file:

**Example 25–1 Template for Silent Initialization File**

```
[General]
Server=
User=
Password=
Direct_Port=
Direct_Secure=
Discover_Port=
HTTPS_PORT=
HTTPS_Secure=
Proxy_Type=
Proxy=
```

The following table describes the silent initialization file's parameters:

**Table 25–2 Initialization File Parameters**

Initialization File Parameter	Description
Server	Oracle Beehive server address
User	Oracle Beehive user account
Password	User's password  <b>Note:</b> It is recommended that you use the option /s UI=1 instead of specifying the password with this parameter. With the /s UI=1 option, the connection dialog is presented in which all the controls are disabled and only the password field is enabled.

**Table 25–2 (Cont.) Initialization File Parameters**

Initialization File Parameter	Description
Direct_Port	Valid port number to be used to establish a direct connection
Direct_Secure	Boolean value; specifies whether to use a direct connection (TRUE) or not (FALSE)
Discover_Port	Valid port number used to connect to the DMS on the server to obtain the settings for direct and HTTPS connections  By using this parameter, you do not need to specify any of the following parameters: Direct_Port, Direct_Secure, HTTPS_PORT, and HTTPS_Secure.
HTTPS_PORT	Valid port number to be used to establish an HTTPS connection
HTTPS_Secure	Boolean value; specifies whether to use an HTTPS connection (TRUE) or not (FALSE)
Proxy_Type	Specifies the proxy type. It may have a value of 0, 1, or 2: <ul style="list-style-type: none"> <li>0: Automatic proxy setting</li> <li>1: Use Web browser's proxy settings</li> <li>2: Manual proxy configuration as specified in the Proxy parameter</li> </ul>
Proxy	Manual proxy configuration (for example, myproxy.us.oracle.com:1234)

## Updating and Configuring Oracle Beehive Extensions for Outlook Through DMS

You may update the version of Oracle Beehive Extensions for Outlook in the DMS repository. Afterwards, if Oracle Beehive Extensions for Outlook is running on an end user's computer, it will automatically prompt the end user to download and install updates from the DMS repository.

In addition, end users may also manually search for updates by selecting **About Oracle Beehive Extensions for Outlook** from the **Help** menu and clicking **Check for Updates**.

You may also apply customized configuration settings to end users' computers from the DMS.

The following topics are covered in this section:

- [Updating Oracle Beehive Extensions for Outlook in DMS Repository](#)
- [Updating Language Pack for Oracle Beehive Extensions for Outlook](#)
- [Applying Configuration Settings from DMS](#)

### Updating Oracle Beehive Extensions for Outlook in DMS Repository

Follow these steps to update the version of Oracle Beehive Extensions for Outlook in the DMS repository:

1. Ensure the application package for the updated version of Oracle Beehive Extensions for Outlook is accessible to the Oracle Beehive server.
2. Upload the updated version of Oracle Beehive Extensions for Outlook to the DMS repository by running the following command:

```
beectl upload_client_application
```



---

```
--file <Path to Oracle Beehive Extensions for Outlook zip file>
```

---

**Note:** This command will upload the new version of the Oracle Beehive Extensions for Outlook application to the DMS repository and will make it available for those who are already provisioned to it.

If you have not provisioned Oracle Beehive Extensions for Outlook, run the provisioning commands (such as `beectl add_client_application_provisioning`). Refer to "Managing the Device Management Service" in *Oracle Beehive Administrator's Guide* for more information.

---

If Oracle Beehive Extensions for Outlook is running on an end user's computer, it will automatically prompt the end user to download and install updates from the DMS repository. The end users may also manually search for updates by selecting **About Oracle Beehive Extensions for Outlook** from the **Help** menu and clicking **Check for Updates**.

### Updating Language Pack for Oracle Beehive Extensions for Outlook

You may localize the user interface of Oracle Beehive Extensions for Outlook to match Microsoft Outlook languages by updating the language pack with the one found at `<Oracle Beehive 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 update the language pack for Oracle Beehive Extensions for Outlook, upload it to the DMS repository using the `beectl upload_client_application` command. By default, the language pack is uploaded and provisioned to all users. Follow these steps to update the language pack:

1. Ensure the language pack for Oracle Beehive Extensions for Outlook is accessible to the Oracle Beehive server.
2. Upload the language pack for Oracle Beehive Extensions for Outlook to the DMS repository by running the following command:

```
beectl upload_client_application
--file <Path to Oracle Beehive Extensions for Outlook language pack>
```

If Oracle Beehive Extensions for Outlook is running on an end user's computer, it will automatically prompt the end user to download and install updates (including language packs) from the DMS repository. End users also may manually search for updates by selecting **About Oracle Beehive Extensions for Outlook** from the **Help** menu and clicking **Check for Updates**.

---

**Note:** The language installed by the Oracle Beehive Extensions for Outlook language pack will automatically match the Outlook language installed by the end user, provided that Oracle Beehive Extensions for Outlook supports the language. If the end user has installed Microsoft Outlook in an unsupported language, Oracle Beehive Extensions for Outlook will default to English.

---

## Applying Configuration Settings from DMS

When deploying Oracle Beehive Extensions for Outlook from DMS, you may apply customized configuration settings to end users' computers. You may associate the configuration settings to a specific version or all versions of the provisioned Oracle Beehive Extensions for Outlook software. Configuration settings associated with all versions are persistent and are always available regardless of the version of Oracle Beehive Extensions for Outlook provisioned to the user. However, configuration settings associated with a specific version are only applied when users are provisioned to that specific version of the software.

---

**Note:** When settings are applied to the specific version level, if the user is later provisioned to a newer software release, the configurations settings are no longer applied.

---

This section covers the following topics:

- [Associating Configuration Settings to All Versions of Application](#)
- [Associating Configuration Settings to Specific Version of Application](#)

## Associating Configuration Settings to All Versions of Application

Follow these steps to apply configuration settings at the application level from the DMS:

1. Create an application package. This is a zip file that contains the following two files:
  - `metadata.xml`: This is an XML provisioning file that includes all the setting parameters you want to apply. Refer to [Example 25–2, "Sample metadata.xml XML Provisioning File"](#) for an example.
  - `readme.txt`: This is simply a text file. You may place any content you want in this file.
2. Upload the application package into the DMS repository with the following command:

```
beectl upload_client_application --file <path to application package zip file>
```
3. Obtain your enterprise EID by running the following command:

```
beectl list_enterprises --entity_format id
```
4. Obtain your application ID by running the command `beectl list_client_applications` and looking for "Application Settings for Oracle Beehive Extensions for Outlook" in the generated list. This is what is in the sample `metadata.xml` provisioning file.
5. Provision the new application by running the following command:

```
beectl add_client_application_provisioning
--community <enterprise EID obtained in step 3>
--application <application ID obtained in step 4>
```

### Example 25–2 Sample metadata.xml XML Provisioning File

The following is a sample XML provision file. It sets the parameter

FILE:profile.ini:GENERAL:GAL-DOWNLOAD-UDS-ON-DEMAND-CACHING-ONLY to FALSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
 <property>
 <name>Application Settings for Oracle Beehive Extensions for Outlook </name>
 <description>
 Oracle Beehive Extensions for Outlook Package Settings
 </description>
 <os>Windows_NT</os>
 <processor>x86</processor>
 <deviceclass>OUTLOOK-WORKSTATION</deviceclass>
 <language>ALL</language>
 <version>1</version>
 <versionnumber>1</versionnumber>
 <patchsetnumber>0</patchsetnumber>
 <vendor>Oracle</vendor>
 </property>
 <modules>
 <module>
 <name>readme.txt</name>
 <src>.</src>
 <dest></dest>
 <contenttype>application/octet-stream</contenttype>
 </module>
 </modules>
 <configuration>
 <param name="FILE:profile.ini:GENERAL:GAL-DOWNLOAD-UDS-ON-DEMAND-CACHING-ONLY"
value="FALSE"></param>
 </configuration>
</application>
```

### Associating Configuration Settings to Specific Version of Application

Follow these steps to apply configuration settings at the specific version level from the DMS:

1. Create an XML provisioning file that includes all the setting parameters you want to apply. For more information, refer to ["Customizing Installation Settings"](#).
2. Obtain your Oracle Beehive Extensions for Outlook identifier by running the `beectl list_client_applications` command and locating the identifier from the generated list.
3. Obtain your Oracle Beehive Extensions for Outlook version identifier by running the following command:

```
beectl list_client_application_versions --application <identifier>
```

*<identifier>* is the Oracle Beehive Extensions for Outlook identifier you obtained in the previous step.

4. Include the obtained Oracle Beehive Extensions 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 running 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 running the following command:

```
beectl add_client_application_provisioning
--community <enterprise ID>
--configuration <application configuration ID>
```

**Customizing Installation Settings** To customize installation settings, create an XML provisioning file with your customized settings as described in ["Creating XML Provisioning File to Customize Installation Settings"](#). Once you have uploaded the provisioning file to DMS, you may apply your customized settings to your end users' computers as described in ["Applying Configuration Settings from DMS"](#).

This section describes how to create an XML provisioning file and the configuration settings you may customize. It covers the following topics:

- [Creating XML Provisioning File to Customize Installation Settings](#)
- [Controlling New Global Address List Settings from DNS](#)
- [Modifying Registry Settings](#)
- [Modifying INI File Settings](#)
- [Enforcing Upgrades](#)

**Creating XML Provisioning File to Customize Installation Settings** You may control general installation 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 may modify the default public properties in your provisioning file by changing the value parameter associated with any of the `INSTALL:<X>` Property nodes.

The following is a sample XML provisioning file. To upload a provisioning file to DMS and then apply its settings to end users' computers, refer to ["Applying Configuration Settings from DMS"](#).

**Example 25–3 Sample XML Provisioning File**

```
<?xml version="1.0" encoding="UTF-8"?>
<ClientAppConfig>
 <Name>Oracle Beehive Extensions for Outlook-test2</Name>
 <Version
 CollabId="6EAB:574C:capv:4672A63200D4B462E040548C17821C2E00000014C0A"/>
 <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;000000000000000;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>

```

The following table describes the public properties you may specify in your XML provisioning file:

**Table 25–3 Public Properties**

Property	Function
USERNAME	Sets the name of the registered Oracle Beehive Extensions for Outlook end user that appears in the <b>About</b> box.
COMPANYNAME	Sets the name of the registered company that appears in the <b>About</b> box.
INSTALLDIR	Sets the location of the Oracle Beehive Extensions for Outlook installation.
CREATE_PROFILE	Determines if an Oracle Beehive Extensions for Outlook profile should be created on the user's computer during installation. This parameter may be assigned one of the following values: <ul style="list-style-type: none"> <li>■ 0: No profile is created.</li> <li>■ 1: Force-add a profile with the specified name.</li> <li>■ 2: Create a profile only if the end user is running a fresh install of Oracle Beehive Extensions for Outlook. This is the default value for the parameter.</li> <li>■ 3: Create a profile only if no profile with the same name exists on the end user's computer.</li> </ul>
PROFILE_NAME	Changes the name of the profile created during installation of Oracle Beehive Extensions for Outlook. The default profile name is Beehive.

---

**Note:** The values you set for the first three parameters in the table only apply when you are installing Oracle Beehive Extensions for Outlook for the first time and do not apply to product upgrades.

---

**Controlling New Global Address List Settings from DNS** A new feature has been introduced in Oracle Beehive Extensions for Outlook that controls what is initially downloaded in the Global Address List (GAL). By default, the Global Address List of any user with a new or migrated profile initially contains only the logged in user's manager, peers, direct reports, and groups. Additional contacts are downloaded and added to the local Global Address List on an ad-hoc basis whenever the user performs an action that requires a contact lookup against the server (for example, creating or receiving e-mail and creating or receiving meetings). This is referred as the Smart GAL mode.

To disable this behavior, the administrator can push the following configuration parameter through the DMS:

- Name: FILE:profile.ini:GENERAL:GAL-DOWNLOAD-UDS-ON-DEMAND-CACHING-ONLY
- Value: FALSE

Provision this parameter to users by following the steps described in ["Applying Configuration Settings from DMS"](#).

#### **Advantages of Smart GAL mode**

- Users do not have to wait till the entire GAL is downloaded.
- The initial GAL size is small and it does not take long to download and construct the initial list.
- The GAL grows dynamically and adds entries in the local list incrementally as the user communicates with other users and groups.

- Adding new contacts and groups is easy and automatic.

### Limitations of Smart GAL mode

- The initial local GAL does not contain the full list as on the server.
- If needed, users have to initiate the download of the full GAL manually from the menu options.
- Server lookup for new contacts and groups is necessary when they are not in the local GAL.

**Modifying Registry Settings** You may modify the registry information on end-user computers by changing the values for certain Property nodes in your provisioning file. Modifying registry information enables 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.

---

- **Adding Key or Registry Value:** To add a key or registry value, use the following in your provisioning file:

```
<Property
 name="REG:ADD:{HIVE}\{keypath}:{Value Name}:{DataType}"
 value="{Value Data}">
</Property>
```

- The value for *{HIVE}* may be either HKLM or HKCU, which are the two locations where user registry information may be stored and which correspond to HKEY\_LOCAL\_MACHINE and HKEY\_CURRENT\_USER in the registry.
- The value for *{DataType}* may 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}*.

- **Deleting Key or Registry Value:** To delete a key or registry value, use the following in your provisioning file:

```
<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.

**Modifying INI File Settings** Modify Oracle Beehive Extensions for Outlook INI settings by including a property with the following structure:

```
<Property
 name="FILE:Beehive.ini:{Section Name}:{Property Name}"
 value="{Value Data}">
</Property>
```

**Enforcing Upgrades** You may force end users to upgrade to a new version of Oracle Beehive Extensions for Outlook that you have uploaded to the DMS. End users will not be able to use Oracle Beehive Extensions for Outlook until they upgrade.

- **Forcing End Users to Upgrade to New Version of Oracle Beehive Extensions for Outlook:** In your XML provisioning file, add a property with the following structure:

```
<Property>
 name="INSTALL:ForceClientVersion:{comparison operation}"
 value="{build number}">
</Property>
```

*{build number}* is the Oracle Beehive Extensions for Outlook build number, which will be used for the comparison operation.

- **Allowing End Users to Use Older Version of Oracle Beehive Extensions for Outlook Until Certain Date:** Optionally, you may allow end users to continue using their old version of Oracle Beehive Extensions for Outlook for a specific period of time or for a fixed number of logins to Oracle Beehive.

In your XML provisioning file, add a property with the following structure:

```
<Property>
 name="INSTALL:graceperioditerations"
 value="{number of logins}">
</Property>
```

*{number of logins}* is the number of times end users are allowed to login to Oracle Beehive Extensions for Outlook before they are forced to upgrade.

## Uninstalling and Downgrading Oracle Beehive Extensions for Outlook (DMS Process)

Use the following methods to uninstall or downgrade Oracle Beehive Extensions for Outlook.

### Uninstalling Oracle Beehive Extensions for Outlook

End users may remove Oracle Beehive Extensions for Outlook from their computers by selecting **Add/Remove Programs** from the Control Panel, selecting **Oracle Beehive Extensions for Outlook** from the list, and clicking **Remove**.

### Downgrading Oracle Beehive Extensions for Outlook

By default, Oracle Beehive Extensions for Outlook will only detect updates if the server version is newer than the currently installed version. If you want to allow end users to install downgraded versions of Oracle Beehive Extensions for Outlook as updates, set the value of the `INSTALL:AllowDowngrade` Property node to `TRUE` in your provisioning file.

## Desktop Based Installation (Non-DMS Process)

It is recommended to install Oracle Beehive Extensions for Outlook using Oracle's Device Management Service (DMS). However, custom environments and lockdown systems with limited administrative privileges may require custom or alternative methods of installation, deployment, and configuration that is non-DMS based.

This section covers the following topics:

- [Installing Oracle Beehive Extensions for Outlook Using Executable File](#)
- [Configuring Profile for Lockdown Systems Using MSI Package](#)
- [Configuring Oracle Beehive Extensions for Outlook Using Configuration Wizard](#)



- [Upgrading Oracle Beehive Extensions for Outlook Installation](#)
- [Updating Language Pack for Oracle Beehive Extensions for Outlook \(Non-DMS\)](#)
- [Uninstalling and Downgrading Oracle Beehive Extensions for Outlook \(Non-DMS Process\)](#)

## Installing Oracle Beehive Extensions for Outlook Using Executable File

You may give end users direct access to the Oracle Beehive Extensions for Outlook installer. The end users install the product by double-clicking the installer and following the on-screen instructions.

Retrieve the installer outlook\_extensions\_setup.exe from <Oracle Beehive home>/beehive/bootstrap/obio/setup.

### Installing Oracle Beehive Extensions for Outlook in Silent Mode

To install Oracle Beehive Extensions for Outlook in silent mode, run the following command:

```
outlook_extensions_setup.exe /s UI=<installer UI option>
```

<installer UI option> is one of values in the following table:

**Table 25–4 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 Extensions for Outlook in silent mode, use the msiexec.exe utility and specify the Oracle Beehive Extensions for Outlook MSI product code as follows:

```
msiexec.exe /x {A634A2AF-2495-4F36-B88F-0B24B84A183C} /qn
```

## Configuring Profile for Lockdown Systems Using MSI Package

You may create customized Oracle Beehive Extensions for Outlook profiles for locked down users. This involves running the Oracle Beehive Extensions for Outlook MSI installer package and specifying the profile parameters as in the command line as public properties. This method assumes that Oracle Beehive Extensions for Outlook was previously installed by an administrator. It does not reinstall Oracle Beehive Extensions for Outlook; it only recreates the Oracle Beehive Extensions for Outlook profile for the locked down user.

Ideally, run the Oracle Beehive Extensions for Outlook MSI installation package on the lockdown system in the locked down user's Windows login script or add an entry to run it, in the runOnce key in the registry.

Run the Oracle Beehive Extensions for Outlook MSI installation package with the following command line options (line breaks have been inserted for clarity):

```
msiexec
-i {MSI_PRODUCT_CODE}
```

```

REINSTALL=ALL
REINSTALLMODE=u
CREATE_PROFILE=3
PROFILE_NAME=MyProfileName {List of public properties with their values}
/qn

```

The following is an example of running the MSI package:

```

msiexec -i {A634A2AF-2495-4F36-B88F-0B24B84A183C}
REINSTALL=ALL
REINSTALLMODE=u
CREATE_PROFILE=3
PROFILE_NAME=MyprofileName
BEEHIVE_SERVER=staqm05.us.oracle.com
BEEHIVE_USER=ferasa
BEEHIVE_PORT_DIRECT=12345
BEEHIVE_PORT_HTTPS=5533
BEEHIVE_TIMEOUT=40000
BEEHIVE_PROXY_TYPE=0
BEEHIVE_DATA_DIRECT=65536
BEEHIVE_DATA_HTTPS=257
/qn

```

The following table describes the public properties you may specify:

**Table 25–5 MSI Package Public Properties**

Property	Description
CREATE_PROFILE	Determines if an Oracle Beehive Extensions for Outlook profile should be created on the user's computer during installation. This parameter may be assigned one of the following values: <ul style="list-style-type: none"> <li>0: No profile is created</li> <li>1: Force-add a profile with the specified name</li> <li>2: Default value; create a profile only if the end user is running a fresh install of Oracle Beehive Extensions for Outlook</li> <li>3: Create a profile only if no profile with the same name exists on the end user's computer</li> </ul>
PROFILE_NAME	Changes the name of the profile created during installation of Oracle Beehive Extensions for Outlook. The default profile name is Beehive.
BEEHIVE_SERVER	Oracle Beehive server address.
BEEHIVE_USER	Oracle Beehive user account.
BEEHIVE_PORT_DIRECT	Valid port number to be used to establish a direct connection.
BEEHIVE_PORT_HTTPS	Valid port number to be used to establish an HTTPS connection.
BEEHIVE_TIMEOUT	The Timeout duration before reporting failure to connect. (milliseconds)
BEEHIVE_PROXY_TYPE	Specifies the proxy type: <ul style="list-style-type: none"> <li>1: Automatic proxy setting</li> <li>2: Manual proxy configuration</li> </ul>
BEEHIVE_DATA_DIRECT	<ul style="list-style-type: none"> <li>256: Disable the direct connection</li> <li>65536: Enable the Direct connection/ No SSL</li> <li>65792: Enable the Direct connection/ With SSL</li> </ul>

**Table 25–5 (Cont.) MSI Package Public Properties**

Property	Description
BEEHIVE_DATA_HTTPS	<ul style="list-style-type: none"> <li>■ 257: Disable the secure HTTP connection</li> <li>■ 65537: Enable the secure HTTP connection / No secure redirection</li> <li>■ 65793: Enable the HTTP connection/ With secure redirection</li> </ul>

## Configuring Oracle Beehive Extensions for Outlook Using Configuration Wizard

The `obioprofwiz.exe` utility, or configuration wizard, is a standalone application that is useful for both administrators and end users because it integrates the creation and configuration of PRF files and profiles. (A PRF file is a text file that Microsoft Outlook uses to generate a profile.) The configuration wizard reads information from an existing PRF file and creates a new profile or modifies an existing profile with the appropriate settings. Start this utility from the command prompt or Windows Explorer. Find `obioprofwiz.exe` in `C:\Program Files\Oracle\Outlook Extensions`.

You may perform the following with the configuration wizard:

- Start the configuration wizard in interactive mode.
- Start the configuration wizard in silent mode; you may use an existing PRF file with all the required settings to create and configure the first profile for users.
- As an administrator, generate the PRF files to save updated settings.
- As an administrator, back up Oracle Beehive Extensions for Outlook settings from an existing profile and restore these settings later.
- As an administrator, modify profile settings in the PRF file generated by the configuration wizard.

Refer to "[Configuration Wizard Modes](#)" for more detailed information about the different modes of the configuration wizard.

---

**Note:** While configuring a profile, it only configures Oracle Beehive Extensions for Outlook settings. You may run only one instance of the configuration wizard at a time. If you try to launch the configuration wizard a second time, the instance previously started will be displayed in the foreground.

---

### Command Syntax

- `obioprofwiz.exe /s <full path to PRF file>`
- `obioprofwiz /d`
- `obioprofwiz /p`

### Options

**Table 25–6 Configuration Wizard Command Line Options**

Option	Description
/d	Specifies that the <b>Set as Default Profile</b> option should be selected by default on the final screen of the configuration wizard.

**Table 25–6 (Cont.) Configuration Wizard Command Line Options**

Option	Description
/p	Runs the configuration wizard in interactive profile creation mode and creates a PRF file.
/s <full path to PRF file>	Runs the configuration wizard in silent profile creation mode. Specify the full path to the PRF file after this option.

---

**Note:** Although the options in the preceding table are shown as lower-case letters, these command-line options are not case-sensitive.

You may enter a slash (/) or a minus sign (-) before an option at the command line.

---

### Examples

- Create a profile interactively:

```
obioprofwiz.exe
```

- Run the configuration wizard in silent profile creation mode:

```
obioprofwiz.exe /s "C:\Program Files\Oracle\Outlook Extentions\source.prf"
```

- Ensure that the **Set as Default Profile** option is selected in the **Congratulations** screen of the configuration wizard.

```
obioprofwiz.exe /d
```

### Configuration Wizard Modes

This section describes the following modes in which you can run the configuration wizard:

- [Interactive Profile Creation and Configuration Mode](#)
- [Interactive PRF File Creation Mode](#)
- [Silent Profile Creation Mode](#)

**Interactive Profile Creation and Configuration Mode** The interactive profile creation and configuration mode is the default mode. Use this mode to specify settings and create or modify a profile. Configure an existing profile or create a new profile by using the **Profile Settings** screen of the configuration wizard. However, you may configure only one profile in one session. You may also remove an existing profile on the **Profile Settings** screen.

1. Close Microsoft Outlook before starting the configuration wizard.
2. Launch the wizard by double-clicking the Oracle Beehive Extensions for Outlook configuration wizard, `obioprofwiz.exe`, from the installation directory.
3. From the **Welcome** screen, click **Next**.
4. If you want to create a new profile, select **New** and enter a name for the new profile. If you want to configure an existing profile, select it from the **Profiles** list. Click **Next**.
5. You may view or modify settings for the selected user profile in the **Service Settings** screen. You must fill all non-password fields to proceed to the next

screen. To configure advanced settings, such as resetting your local mailbox cache, setting a folder refresh rate for incoming mail, or selecting a refresh option for the Global Address Book, click **More Settings**. Click **Next** to proceed.

---

**Note:** If Microsoft Outlook is open and you are using the same profile while you are configuring profile settings, the fields on the **Service Settings** screen are disabled.

---

- When the **Congratulations** screen is displayed, verify that **Set as Default Profile** is selected. The Outlook Address Book service is added to a new profile by default.

**Interactive PRF File Creation Mode** As an administrator, you may run the `obioprofwiz.exe` utility in interactive PRF file creation mode. In this mode, you may create a PRF file by starting with a blank template or by importing information from an existing profile or PRF file.

By default, the PRF file created is a blank template. For security reasons, password fields are not saved in the resulting PRF file if information is imported from an existing profile. The final PRF file contains user profile information and settings information, which may be used by the configuration wizard or by the `NewProf.exe` utility.

- Close Microsoft Outlook before starting the configuration wizard.
- Run the configuration wizard with the `/p` option at the command line. For example:

```
obioprofwiz.exe /p
```

- In the **Source Template in the PRF Configuration Settings** screen, select one of the following options for the PRF file:
  - Empty PRF File:** If you select this option as the source template, you must manually specify the settings.
  - Existing PRF File:** If you select this option as the source template, initial settings information is imported from the source PRF file.
  - MAPI Profile:** If you select this option as the source template, initial settings information is imported from the source profile.

---

**Note:** If you import information from an existing PRF file or MAPI profile, Microsoft Outlook address book information is added to the Service List section of the output PRF file if the existing PRF file or MAPI profile already contains address book information or the Outlook Address Book service, respectively.

---

- Enter the name and location of the new PRF file in the **Destination File** field.
- In the **Profile Options** screen, configure the settings for the General section of the new PRF file. The following table shows the relationship between the fields in the **Profile Options** screen and the parameters in the General section of the PRF file:

**Table 25–7 Relationship Between Fields in the Profile Options Screen and Settings in the General Section of the PRF File**

Field in Profile Options Screen	Parameter in General Section of PRF File
Profile Name	ProfileName

**Table 25–7 (Cont.) Relationship Between Fields in the Profile Options Screen and Settings in the General Section of the PRF File**

Field in Profile Options Screen	Parameter in General Section of PRF File
Default Store	DefaultStore
Use this profile as the default MAPI profile	DefaultProfile=Yes/No
Overwrite existing profile with the same name	OverwriteProfile=Yes/No

6. View or modify Oracle Beehive Extensions for Outlook settings for the selected user profile in the **Service Settings** screen. All non-password fields must be filled to proceed to the next screen. Click **Next** to proceed.

To configure advanced settings, such as resetting your local mailbox cache, setting a folder refresh rate for incoming mail, or selecting a refresh option for the Global Address Book, click **More Settings**. The Oracle Beehive Extensions for Outlook settings information will be saved to the new PRF file.

7. If **View PRF File** is selected when the **Congratulations** screen is displayed, the wizard displays the newly created PRF file. You may then verify the information or modify parameters in the PRF file.

**Silent Profile Creation Mode** In this mode, you may use an existing PRF file as a template to create a corresponding profile. When installing Oracle Beehive Extensions for Outlook in silent mode, you may start the configuration wizard after the installation is complete. The configuration wizard reads information from an existing PRF file and creates a new profile or modifies an existing profile with the appropriate settings.

---

**Note:** Oracle recommends that you use a PRF file generated by the Oracle Beehive Extensions for Outlook configuration wizard. You may configure most of the parameters in the PRF file with the configuration wizard in the interactive PRF file creation mode. If you modify a value manually, then you must verify that the new value falls within the valid range of values. To verify that you are within the valid range of values, consult the PRF file distributed with the installation package.

---

The Outlook Address Book service is imported into the profile if it is listed in the Service List section. Limited support is also available for some other types of message services.

When Oracle Beehive Extensions for Outlook is configured in silent mode, a log file is created to record success and error information. This log file is stored in a temporary folder on the computer with a path and file name similar to C:\Documents and Settings\user\Local Settings\Temp\obioConfigWizardLog\obioConfigWizard.log, where user is the login name of the currently logged in Windows user.

---

**Note:** If you are configuring Oracle Beehive Extensions for Outlook in silent mode, then you must check the log file to verify the results. The log file contains a record of errors, as well as successful imports of information into the profile.

The configuration wizard may write some information to the log file in interactive profile creation and configuration mode and interactive PRF file creation mode. Usually, high-level warning messages are displayed on the screen, while low-level technical information is recorded in the log file.

---

## Upgrading Oracle Beehive Extensions for Outlook Installation

When you provision a new version of Oracle Beehive Extensions for Outlook in the DMS repository, end users will be notified about the new update and will be able to download and install it.

If administrators prefer to upgrade to a newer version of Oracle Beehive Extensions for Outlook on end users' systems without relying on the DMS upgrade process (because, for example, the computers they want to upgrade are lockdown systems), then they need to deploy the new version of Oracle Beehive Extensions for Outlook before performing any server updates.

Alternatively, you may remove any Oracle Beehive Extensions for Outlook packages from the DMS repository so that end users are never prompted to upgrade to newer versions. Perform the following steps to remove Oracle Beehive Extensions for Outlook packages:

1. Run the following command:

```
beectl list_client_applications
```

2. The previous command should list the following two Oracle Beehive Extensions for Outlook application packages and their identifiers:

- Oracle Beehive Extensions for Outlook
- Oracle Beehive Extensions for Outlook Language Pack

Remove both client applications by running the following command (for each application):

```
beectl delete_client_application
--application <identifier of client application>
```

## Updating Language Pack for Oracle Beehive Extensions for Outlook (Non-DMS)

The default localization of the user interface of OBEO can be changed according to the user preference. This can be done by supplying newer xliif files for the preferred languages that would contain the required changes. If the default language pack is satisfactory, however, then no change will be required and hence no updating of the language pack will be necessary.

Follow the steps below to update the modified language packs:

1. Obtain the default language pack from the following location:  
 <Oracle Beehivehome>/beehive/bootstrap/obio/languagepack
2. Unzip the package.

3. Find the corresponding xliiff file of the language that needs to be modified and make the changes.
4. Place the changed file in a folder.

For example: c:\lab\Updates (This would be the path to xliiff chm files)

5. These modified files can be deployed on the users' machines using one of the following commands:

```
outlook_extension_setup.exe /package_langpack {path_to_a_folder_of_xliiff_chm_files}
```

This command generates a newer installation package with the updated xliiff files packaged inside it. This full installation package is generated in the same folder of the original installation package. It can then be executed normally by the users to install and update their deployed language pack in the deployed location (usually C:\Program Files\Oracle\Outlook Extensions\languages\Updates).

```
outlook_extension_setup.exe /install_langpack {path_to_a_folder_of_xliiff_chm_files}
```

This command does not generate a new installation package. Instead, it copies the modified files directly to its deployed location. It can be used by the user to update the deployed language pack, provided that the user running the installation has access to the folder where the updated files are located (this can be a network location) during the installation time.

## Uninstalling and Downgrading Oracle Beehive Extensions for Outlook (Non-DMS Process)

Use the following methods to uninstall or downgrade Oracle Beehive Extensions for Outlook.

### Uninstalling Oracle Beehive Extensions for Outlook

End users may remove Oracle Beehive Extensions for Outlook from their computers by selecting **Add/Remove Programs** from the Control Panel, selecting Oracle Beehive Extensions for Outlook from the list, and clicking **Remove**.

### Downgrading Oracle Beehive Extensions for Outlook

If you want to install downgraded versions of Oracle Beehive Extensions for Outlook on end users' systems, then simply run the installer older Oracle Beehive Extensions for Outlook version.

## Installing Oracle Beehive Extensions for Outlook Using Active Directory

If your end users are running systems on Windows XP or Windows Vista that are locked down, they may install Oracle Beehive Extensions for Outlook using an Active Directory group policy. To perform an Active Directory installation, the domain administrator must create a package for Oracle Beehive Extensions 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.

This section covers the following topics:



- [Creating MSI installation package for Oracle Beehive Extensions for Outlook](#)
- [Deploying Oracle Beehive Extensions for Outlook using Active Directory Group Policy](#)
- [Verifying that Oracle Beehive Extensions for Outlook Is Available for Installation](#)

## Creating MSI installation package for Oracle Beehive Extensions for Outlook

Active Directory only recognizes software packaged in the MSI file format, which contains information about the application setup and installation. Follow these steps to create an MSI installation package for Oracle Beehive Extensions for Outlook:

1. Extract the outlook\_extensions\_application.msi file from the Oracle Beehive Extensions for Outlook package using the outlook\_extensions\_setup.exe /export type=client <extracted folder path> command.
2. Extract the prerequisite.msi file from the Oracle Beehive Extensions for Outlook package using the outlook\_extensions\_setup.exe /export type=system <extracted folder path> command.

## Deploying Oracle Beehive Extensions for Outlook using Active Directory Group Policy

Before proceeding, ensure you have extracted the MSI files for Oracle Beehive Extensions for Outlook to an accessible location. Follow these steps to deploy Oracle Beehive Extensions for Outlook using an 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 Extensions 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 Extensions 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 Extensions 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\_extensions\_application.msi file.

---

**Note:** For more information on how to obtain MSI files for Oracle Beehive Extensions for Outlook, see [Creating MSI installation package for Oracle Beehive Extensions for Outlook](#).

---

15. In the **Properties** dialog box, click **OK**.
16. Exit the **Active Directory Users and Computers** console.

## Verifying that Oracle Beehive Extensions for Outlook Is Available for Installation

To verify that Oracle Beehive Extensions for Outlook is available to end users for installation, restart any computer in the domain and make sure that Oracle Beehive Extensions for Outlook installation starts when the computer restarts.

## Using Oracle Beehive Extensions for Outlook as Profile Migration Tool

You may also use the Oracle Beehive Extensions for Outlook downloader as a tool to migrate and integrate some configurations settings into an Oracle Beehive Extensions for Outlook profile. This includes the following options:

- Migrating existing personal folder data files (files with a .pst extension) from previous profiles
- Migrating LDAP settings from previous profiles
- Adding new LDAP settings
- Adding new personal folders data file

Use the `/migrate` switch as follows:

```
outlook_extensions_downloader.exe /migrate C:\migrateinifile.ini
```

If you do not provide an initialization file (such as `C:\migrateinifile.ini` in the previous example), you will be presented with an interactive user interface that enables you to choose the personal folders data files and the LDAP settings from previous profiles to migrate.

Use the following template to create a migration initialization file:

### **Example 25–4   Template for Migration Initialization File**

```
[migrate_psts]
Src=
Dest=
CopyFile=

[migrate_ldaps]
Src=
Dest=

[add_pst]
Path=
Dest=

[add_ldap]
Name=
Host=
Port=
```

Base=  
SSL=  
Dest=

[migrate\_NK2]  
Src=  
Dest=

**Table 25–8 Migration Initialization File Parameters**

Section Name	Parameter Name	Description
[migrate_psts]	Src	Migrates existing personal folder data files (files with a .pst extension) from the specified profile or profiles. It may have one of the following values: <ul style="list-style-type: none"> <li>■ Name of existing profile</li> <li>■ \OCFO\ (Migrates from all Oracle Connector for Outlook profiles that already exist in the system)</li> <li>■ \ALL\ (Migrates from all profiles that already exist in the system)</li> </ul>
	Dest	Existing Oracle Beehive Extensions for Outlook profile to which the migrated settings should be added. This may be the Oracle Beehive profile created with the downloader.
	CopyFile	One of the following values: <ul style="list-style-type: none"> <li>■ TRUE (The migrated PST file will be copied to the offline storage of the destination profile and added to it from the new location)</li> <li>■ FALSE (The destination profile will only have a link to the existing PST file)</li> </ul>
[migrate_ldaps]	Src	Migrates LDAP settings from the specified profile or profiles. It may have one of the following values: <ul style="list-style-type: none"> <li>■ Name of existing profile</li> <li>■ \OCFO\ (Migrates from all Oracle Connector for Outlook profiles that already exist in the system)</li> <li>■ \ALL\ (Migrates from all profiles that already exist in the system)</li> </ul>
	Dest	Existing Oracle Beehive Extensions for Outlook profile to which the migrated settings should be added. This may be the Oracle Beehive profile created with the downloader.
[add_pst]	Path	Adds the settings from a specified PST file. It may have one of the following values: <ul style="list-style-type: none"> <li>■ Absolute path to PST file</li> <li>■ Name of a PST file (its location is determined from the offline storage of the destination profile)</li> </ul>
	Dest	Existing Oracle Beehive Extensions for Outlook profile to which the settings should be added. This may be the Oracle Beehive profile created with the downloader.

**Table 25–8 (Cont.) Migration Initialization File Parameters**

Section Name	Parameter Name	Description
[add_ldap]	Name	Display name of LDAP
	Host	LDAP host name
	Port	LDAP port number
	Base	LDAP base distinguished name (DN)
	SSL	Boolean value; specifies whether SSL is enabled (TRUE) or not (FALSE)
	Dest	Existing Oracle Beehive Extensions for Outlook profile to which the settings should be added. This may be the Oracle Beehive profile created with the downloader.
[migrate_NK2]	Src	Migrates nickname cache from specified profile or profiles. It may have one of the following values: <ul style="list-style-type: none"> <li>■ Name of existing profile</li> <li>■ \OCFO\ (Migrates from all Oracle Connector for Outlook profiles that already exist in the system)</li> <li>■ \ALL\ (Migrates from all Oracle Beehive Extensions for Outlook profiles that already exist in the system)</li> </ul>
	Dest	Existing Oracle Beehive Extensions for Outlook to which the migrated settings should be added. This may be the Oracle Beehive profile created with the downloader.

## Troubleshooting

In certain cases, an application process (such as Windows Search) may lock a file that is part of the Oracle Beehive Extensions for Outlook upgrade, resulting in incomplete installation. The installation process will display a message similar to the following:

The following applications are using files that need to be updated by this setup. Close these applications and click **Retry** to continue.

If this occurs, follow the below mentioned suggestions:

1. Click **Retry** to see if the installation proceeds.
2. If **Retry** does not work, click **Ignore** to see if the installation proceeds.  
If **Ignore** allows the installation to proceed, then you may be prompted to reboot to complete the installation.
3. If **Ignore** does not work, then **Exit** the installation, wait several minutes, and then re-try the installation.
4. If the problem persists despite waiting for a reasonable period, then you will need to disable the Windows Search service, perform the Oracle Beehive Extensions for Outlook installation, and then re-enable the Windows Search service.

---

## Installing Oracle Beehive Extensions for Explorer

This chapter describes how to install Oracle Beehive Extensions for Explorer.

Oracle recommends that you install Oracle Beehive Extensions for Explorer using Oracle Beehive's Device Management Service (DMS). This installation method is recommended for almost all desktops with standard environments and administrative privileges. A desktop based, non-DMS installation method is available for custom environments with limited administrative privileges.

It covers the following topics:

- [System Requirements](#)
- [Device Management Service \(DMS\) Based Installation](#)
- [Desktop Based Installation \(Non-DMS Process\)](#)

### System Requirements

This section describes the software and hardware requirements for Oracle Beehive Extensions for Explorer. For updated list of certified hardware platforms and operating system version, review the certification matrix on the My Oracle Support Web site at the following URL:

<http://support.oracle.com/>

#### Operating System

- Microsoft Windows XP: Home or Professional
- Microsoft Windows 7 (32-bit and 64-bit edition) - Home Premium, Professional, Ultimate or Enterprise
- Microsoft Windows Server 2003
- Microsoft Windows Server 2008
- Microsoft Windows Server 2008 R2

---

---

**Note:** You need administrative privileges to install Oracle Beehive Extensions for Explorer on Windows XP, Windows Server 2003, and Windows Server 2008.

You must install Windows XP Service Pack 2 on Windows XP computers if you intend to use non-English locales with Oracle Beehive Extensions for Explorer. Attempting to run Oracle Beehive Extensions for Explorer 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.

---

---

#### **Disk Space**

- 100 MB minimum
- Hard disk usage varies according to configuration. Custom installation options may require more or less hard disk space. The amount of online data the user accesses may also affect the disk space requirement.

## **Device Management Service (DMS) Based Installation**

Oracle recommends that you install and deploy Oracle Beehive Extensions for Explorer using Oracle Beehive's Device Management Service (DMS). This installation method is recommended for almost all desktops with standard environments and administrative privileges. It is designed for centralized installation and management of auto-updates.

This section covers the following topics:

- [Installing Oracle Beehive Extensions for Explorer Using Remote Downloader](#)
- [Update Oracle Beehive Extensions for Explorer Through DMS](#)
- [Uninstalling and Downgrading Oracle Beehive Extensions for Explorer \(DMS Process\)](#)

## **Installing Oracle Beehive Extensions for Explorer Using Remote Downloader**

You may install Oracle Beehive Extensions for Explorer by distributing a remote downloader to end users. The remote downloader enables end users to download and install the provisioned Oracle Beehive Extensions for Explorer application from the Device Management Service (DMS) repository. By default, the DMS contains a pre-seeded Oracle Beehive Extensions for Explorer application that is provisioned for all end users.

For more information on how to provision applications for specific users or groups of users, refer to "Managing the Device Management Service" in *Oracle Beehive Administrator's Guide*.

#### **Obtaining Remote Downloader**

Obtain the Oracle Beehive Extensions for Explorer remote downloader, `explorer_extensions_downloader.exe` in the directory `<Oracle Beehive home>/beehive/bootstrap/obee/downloader`.

You may also obtain the remote downloader from Oracle Beehive Central, a Web-based client that provides users a central location to download supported clients and set their preferences for Oracle Beehive functionality.

### Pre-Seeding Server Name in 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 want 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

You may pre-seed the port number that Oracle Beehive Extensions for Outlook uses to connect to the server.

For example, if your remote downloader name is `faulkner.oracle.com.exe` and you want to pre-seed the port number in the downloader, rename the file to `faulkner.oracle.com,443.exe`, where 443 is your port number.

---

**Note:** The pre-seeded port number is the port number of the DMS on the server that is used to obtain the settings for direct and HTTPS connections.

---

## Installing Oracle Beehive Extensions for Explorer Using Remote Downloader Using Silent Mode

This option is available for Oracle Beehive Release 1 (1.5.1.2) and later.

You may also use the Oracle Beehive Extensions for Explorer downloader to provide a full, non-interactive deployment option. To do so, use the silent switch, `/s`.

Specify the required connection parameters through the command line or in an initialization file.

**Specifying Connection Parameters Through Command Line** The following is an example of specifying connection parameters through the command line (line breaks have been added for clarity):

```
explorer_extensions_downloader.exe
/server example.com
/user OracleBeehiveUserName
/direct port=5224 secure=true
/https port=443 secure=true
/s UI=1
```

The following table describes the Oracle Beehive Extensions for Explorer downloader's command line options:

**Table 26–1 Oracle Beehive Extensions for Explorer Downloader Options**

Option	Description
<code>/server</code>	Oracle Beehive server address
<code>/user</code>	Oracle Beehive user account
<code>/pass</code>	User's password  <b>Note:</b> It is recommended that you use the option <code>/s UI=1</code> instead of the <code>/pass</code> option. With the <code>/s UI=1</code> option, the connection dialog is presented in which all the controls are disabled and only the password field is enabled.

**Table 26–1 (Cont.) Oracle Beehive Extensions for Explorer Downloader Options**

Option	Description
<code>/direct port=&lt;port number&gt; secure=&lt;true/false&gt;</code>	Specifies the following for direct connection: <ul style="list-style-type: none"> <li>port: Valid port number to be used to establish a direct connection</li> <li>secure: Boolean value; if true, then use a secure connection</li> </ul>
<code>/https port=&lt;port number&gt; secure=&lt;true/false&gt;</code>	Specifies the following for HTTPS connection: <ul style="list-style-type: none"> <li>port: Valid port number to be used to establish an HTTPS connection</li> <li>secure: Boolean value; if true, then use a secure connection</li> </ul>
<code>/create_profile</code>	Determines if an Oracle Beehive Extensions for Explorer profile should be created on the user's computer during installation. This parameter may be assigned one of the following values: <ul style="list-style-type: none"> <li>0: Do not create a profile</li> <li>1: Always create a profile (either for a new install or an upgrade)</li> <li>2: Only create a profile for a new install (default value)</li> <li>3: Only create a profile if there is no profile with the exact name that exists on the user's computer (either for a new install or upgrade)</li> </ul>
<code>/discover port=&lt;port number&gt;</code>	Specifies the port number of the DMS on the server to obtain the settings for direct and HTTPS connections.  By using this option, you do not need to specify either the <code>/direct</code> or <code>/https</code> options.
<code>/profile_name</code>	Changes the name of the profile created during installation of Oracle Beehive Extensions for Outlook. The default profile name is Beehive.
<code>/repair</code>	Downloads and reinstalls the MSI package from the DMS if the installed version on the user's computer is the same version as the MSI package.  If <code>/create_profile</code> option has a value of 1 or 3, then the <code>/repair</code> option is used even if it is not specified.
<code>/launch_explorer</code>	Determines if Windows Explorer is launched after Oracle Beehive Extensions for Explorer is installed. This parameter may be assigned one of the following values: <ul style="list-style-type: none"> <li>0: Do not launch Windows Explorer</li> <li>1: Launch Windows Explorer (default value)</li> </ul>
<code>/s</code>	Use silent mode

**Specifying Connection Parameters in Initialization File** Specify the full path of the initialization file you want to use as follows. In this example, `C:\testSilentDownloader.ini` is the full path of the initialization file:

```
explorer_extensions_downloader.exe /s C:\testSilentDownloader.ini
```



If you use the silent switch with a `UI=1` parameter, the connection dialog will be presented in which all the controls are disabled and only the password field is enabled:

```
explorer_extensions_downloader.exe /s UI=1 C:\testSilentDownloader.ini
```

The downloader will return a value of 1 or 0, which indicates whether the operations performed by it failed (1) or were successful (0). You may use this returned value in a script to determine whether or not the remote downloader was successful.

Use the following template to create a silent initialization file:

**Example 26–1 Template for Silent Initialization File**

```
[General]
Account_Name=
Server=
User=
Password=
Direct_Port=
Direct_Secure=
Direct_Enable=
DiscoverPort=
HTTPS_PORT=
HTTPS_Secure=
HTTPS_Enable=TRUE
Proxy=
Proxy_Type=0
Proxy_URL=
Use_Existing=
Default=
Remember_Pwd=
Timeout=120000
```

The following table describes the silent initialization file's parameters:

**Table 26–2 Initialization File Parameters**

Initialization File Parameter	Description
Account_Name	Changes the name of the profile created during installation of Oracle Beehive Extensions for Outlook. The default profile name is Beehive.
Server	Oracle Beehive server address
User	Oracle Beehive user account
Password	User's password <b>Note:</b> It is recommended that you use the option <code>/s UI=1</code> instead of specifying the password with this parameter. With the <code>/s UI=1</code> option, the connection dialog is presented in which all the controls are disabled and only the password field is enabled.
Direct_Port	Valid port number to be used to establish a direct connection
Direct_Secure	Boolean value; specifies whether to use a direct connection (TRUE) or not (FALSE)
Direct_Enable	Boolean value; specifies whether to enable the direct connection (TRUE) or not (FALSE)

**Table 26–2 (Cont.) Initialization File Parameters**

Initialization File Parameter	Description
Discover_Port	Valid port number used to connect to the DMS on the server to obtain the settings for direct and HTTPS connections  By using this parameter, you do not need to specify any of the following parameters: Direct_Port, Direct_Secure, HTTPS_PORT, and HTTPS_Secure.
HTTPS_PORT	Valid port number to be used to establish an HTTPS connection
HTTPS_Secure	Boolean value; specifies whether to use an HTTPS connection (TRUE) or not (FALSE)
Proxy	Manual proxy configuration (for example, myproxy.us.oracle.com:1234)
Proxy_Type	Specifies the proxy type. It may have a value of 0, 1, or 2: <ul style="list-style-type: none"> <li>0: Automatic proxy setting</li> <li>1: Use Web browser's proxy settings</li> <li>2: Manual proxy configuration as specified in the Proxy parameter</li> </ul>
Proxy_URL	Manual proxy configuration (for example, myproxy.us.oracle.com:1234)
Use_Existing	Boolean value: <ul style="list-style-type: none"> <li>TRUE: Existing account is re-configured based on new settings. Workspace favorites are not reset. No new account is created.</li> <li>FALSE: Existing account is not re-configured. New account is created based on new settings.</li> <li>If no value is specified, existing account is not re-configured. No new account is created.</li> </ul>
Default	Boolean value: Specifies whether the created profile should be the default profile (TRUE) or not (FALSE)
Timeout	Timeout duration, in milliseconds, before reporting failure to connect

## Update Oracle Beehive Extensions for Explorer Through DMS

You may update the version of Oracle Beehive Extensions for Explorer in the DMS repository. Afterwards, if Oracle Beehive Extensions for Explorer is running on an end user's computer, it will automatically prompt the end user to download and install updates from the DMS repository.

In addition, end users may also manually search for updates by selecting **About Oracle Beehive Extensions for Explorer** from the **Beehive** menu and clicking **Check for Updates**.

The following topics are covered in this section:

- [Updating Oracle Beehive Extensions for Explorer in DMS Repository](#)
- [Updating Language Pack for Oracle Beehive Extensions for Explorer](#)

### Updating Oracle Beehive Extensions for Explorer in DMS Repository

Follow these steps to update the version of Oracle Beehive Extensions for Explorer in the DMS repository:

1. Ensure the application package for the updated version of Oracle Beehive Extensions for Explorer is accessible to the Oracle Beehive server.
2. Upload the updated version of Oracle Beehive Extensions for Explorer to the DMS repository by running the following command:

```
beectl upload_client_application
--file <Path to Oracle Beehive Extensions for Explorer zip file>
```

---

**Note:** This command will upload the new version of the Oracle Beehive Extensions for Explorer application to the DMS repository and will make it available for those who are already provisioned to it.

If you have not provisioned Oracle Beehive Extensions for Explorer, run the provisioning commands (such as `beectl add_client_application_provisioning`). Refer to "Managing the Device Management Service" in Oracle Beehive Administrator's Guide for more information.

---

If Oracle Beehive Extensions for Explorer is running on an end user's computer, it will automatically prompt the end user to download and install updates from the DMS repository. End users may also manually search for updates by selecting **About Oracle Beehive Extensions for Explorer** from the **Beehive** menu and clicking **Check for Updates**.

### Updating Language Pack for Oracle Beehive Extensions for Explorer

You may localize the user interface of Oracle Beehive Extensions for Explorer to match Windows Explorer languages by updating the language pack with the one found at `<Oracle Beehive home>/beehive/bootstrap/obee/languagepack`. The following languages are currently supported in the language pack:

- French
- German
- Italian
- Spanish
- Brazilian Portuguese
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

To update the language pack for Oracle Beehive Extensions for Explorer, upload it to the DMS repository using the `beectl upload_client_application` command. By default, the language pack is uploaded and provisioned to all users. Follow these steps to update the language pack:

1. Ensure the language pack for Oracle Beehive Extensions for Explorer is accessible to the Oracle Beehive server.
2. Upload the language pack for Oracle Beehive Extensions for Explorer to the DMS repository by running the following command:

```
beectl upload_client_application
--file <Path to Oracle Beehive Extensions for Explorer language pack>
```

If Oracle Beehive Extensions for Explorer is running on an end user's computer, it will automatically prompt the end user to download and install updates (including language packs) from the DMS repository. End users also may manually search for updates by selecting **About Oracle Beehive Extensions for Explorer** from the **Beehive** menu and clicking **Check for Updates**.

---

**Note:** The language installed by the Oracle Beehive Extensions for Explorer language pack will automatically match Windows Explorer, provided that Oracle Beehive Extensions for Explorer supports the language. If the end user has Windows Explorer in an unsupported language, Oracle Beehive Extensions for Explorer will default to English.

---

## Uninstalling and Downgrading Oracle Beehive Extensions for Explorer (DMS Process)

Use the following methods to uninstall or downgrade Oracle Beehive Extensions for Explorer.

### Uninstalling Oracle Beehive Extensions for Explorer

End users may remove Oracle Beehive Extensions for Explorer from their computers by selecting **Add/Remove Programs** from the Control Panel, selecting **Oracle Beehive Extensions for Explorer** from the list, and clicking **Remove**.

### Downgrading Oracle Beehive Extensions for Explorer

By default, Oracle Beehive Extensions for Explorer will only detect updates if the server version is newer than the currently installed version. If you want to allow end users to install downgraded versions of Oracle Beehive Extensions for Explorer as updates, set the value of the `INSTALL:AllowDowngrade` Property node to `TRUE` in your provisioning file.

## Desktop Based Installation (Non-DMS Process)

It is recommended to install Oracle Beehive Extensions for Explorer using Oracle's Device Management Service (DMS). However, custom environments with limited administrative privileges may require a desktop based method of installation that is non-DMS based.

This section covers the following topics:

- [Installing Oracle Beehive Extensions for Explorer Using Executable File](#)
- [Configuring Oracle Beehive Extensions for Explorer Using MSI Package](#)
- [Upgrading Oracle Beehive Extensions for Explorer Installation](#)
- [Updating Language Pack for Oracle Beehive Extensions for Explorer \(Non-DMS\)](#)
- [Uninstalling and Downgrading Oracle Beehive Extensions for Explorer \(Non-DMS Process\)](#)

## Installing Oracle Beehive Extensions for Explorer Using Executable File

You may give end users direct access to the Oracle Beehive Extensions for Explorer installer. End users install the product by double-clicking the installer and following the on-screen instructions.

Retrieve the installer `explorer_extensions_setup.exe` from `<Oracle Beehive home>/beehive/bootstrap/obee/setup`.

## Configuring Oracle Beehive Extensions for Explorer Using MSI Package

This option is available for Oracle Beehive Release 1 (1.5.1.2) and later.

The following table describes the public properties you may specify:

**Table 26–3 MSI Package Public Properties**

Property	Description
CREATE_PROFILE	<p>Determines if an Oracle Beehive Extensions for Explorer profile should be created on the user's computer during installation. This parameter may be assigned one of the following values:</p> <ul style="list-style-type: none"> <li>0: Do not create a profile</li> <li>1: Always create a profile (either for a new install or an upgrade)</li> <li>2: Only create a profile for a new install (default value)</li> <li>3: Only create a profile if there is no profile with the exact name that exists on the user's computer (either for a new install or upgrade)</li> </ul>
EXISTING_PROFILE_OPTION	<p>Boolean value:</p> <ul style="list-style-type: none"> <li>TRUE: Existing account is re-configured based on new settings. Workspace favorites must not be reset. No new account is created.</li> <li>FALSE: Existing account must not be re-configured. New account is created based on new settings.</li> <li>If no value is specified, existing account must not be re-configured. No new account is created.</li> </ul> <p>If CREATE_PROFILE=2, EXISTING_PROFILE_OPTION is not used, and the Use_Existing option in the configuration file is not used, then the value of this property is TRUE.</p>
PROFILE_NAME	<p>Changes the name of the profile created during installation of Oracle Beehive Extensions for Outlook. The default profile name is Beehive.</p>
LAUNCH_EXPLORER	<p>Determines if Windows Explorer is launched after Oracle Beehive Extensions for Explorer is installed. This parameter may be assigned one of the following values:</p> <ul style="list-style-type: none"> <li>0: Do not launch Windows Explorer</li> <li>1: Launch Windows Explorer (default value)</li> </ul>

### Custom MSI-based Deployment and Configuration

This section describes how to deploy custom OBEE MSI package.

Create a silent `obeeconfig.ini` file and specify the required connection parameters in the initialization file.

Use the following template to create a silent initialization file:

#### Example 26–2 Template for Silent Initialization File

```
[General]
Account_Name=Beehive
Server=yourservername.com
User=username@yourcompany.com
```

```
Password=
Direct_Port=5224
Direct_Secure=TRUE
Direct_Enable=TRUE
HTTPS_PORT=443
HTTPS_Secure=TRUE
HTTPS_Enable=TRUE
Proxy=
Proxy_Type=0
Proxy_URL=
Use_Existing=TRUE
Discover_Port=443
Discover_Enable=TRUE
Default=
Remember_Pwd=
Timeout=60000
```

For information on the description of initialization file's parameters, refer [Table 26–2, "Initialization File Parameters"](#).

To configure OBEE, after installing as an admin user, follow the below mentioned steps:

1. Install Oracle Beehive Extensions for Explorer by running the following command. This requires admin privilege.

```
msiexec.exe -i "C:\testObeeMsi\explorer_extensions_application.msi"
```

2. Place the obeeconfig.ini file in the installation directory, or in a network location where users have access. You do not need to make the ini file read-only.
3. Configure the OBEE profile according to the settings in the ini file. This step does not require admin privilege and will repeat for each user.

- a. Copy the obeeconfig.ini file to a temp location of the logged in user (You do not need to make the ini file read-only).
- b. Add a new String value called ConfigPath and populate it with the full path to the obeeconfig.ini file under the following registry path:

```
HKEY_CURRENT_USER\Software\Oracle\Explorer Extensions
```

For example:

```
ConfigPath C:\testObeeMsi\obeeconfig.ini
```

- c. Insert the username in the ini file from the temp location.

---

**Note:** As part of the post installation/setup cleanup process, the deployed ini file will be consumed and removed when OBEE is first launched on your machines. As the process works with a copy from the temp location, the ini file does not have to be read-only and can be invoked for the user name insertion process and configuration.

---

## Upgrading Oracle Beehive Extensions for Explorer Installation

When you provision a new version of Oracle Beehive Extensions for Explorer in the DMS repository, end users will be notified about the new update and will be able to download and install it.

If administrators prefer to upgrade to a newer version of Oracle Beehive Extensions for Explorer on end users' systems without relying on the DMS upgrade process, then

they need to deploy the new version of Oracle Beehive Extensions for Explorer before performing any server updates.

Alternatively, you may remove any Oracle Beehive Extensions for Explorer packages from the DMS repository so that end users are never prompted to upgrade to newer versions. Perform the following steps to remove Oracle Beehive Extensions for Explorer packages:

1. Run the following command:

```
beectl list_client_applications
```

2. The previous command will return a list of the applications installed in the DMS repository and their identifiers. Find the listings pertaining to OBEE, you may see any of the following Oracle Beehive Extensions for Explorer application packages and their identifiers:

- Oracle Beehive Extensions for Explorer
- Oracle Beehive Extensions for Explorer Downloader
- Oracle Beehive Extensions for Explorer Language Pack

Remove any of these client applications found by running the following command (for each application):

```
beectl delete_client_application
--application <identifier of client application>
```

## Updating Language Pack for Oracle Beehive Extensions for Explorer (Non-DMS)

The default localization of the user interface of OBEE can be changed according to the user preference. This can be done by supplying newer xliif files for the preferred languages that would contain the required changes. If the default language pack is satisfactory, however, then no change will be required and hence no updating of the language pack will be necessary.

Follow the steps below to update the modified language packs:

1. Obtain the default language pack from the following location:  
`<Oracle Beehivehome>/beehive/bootstrap/obee/languagepack`
2. Unzip the package.
3. Find the corresponding xliif file of the language that needs to be modified and make the changes.
4. Place the changed file in a folder.

For example: c:\lab\Updates (This would be the path to xliif chm files)

5. These modified files can be deployed on the users' machines using one of the following commands:

```
explorer_extension_setup.exe /package_langpack {path_to_a_folder_of_xliif_chm_files}
```

This command generates a newer installation package with the updated xliif files packaged inside it. This full installation package is generated in the same folder of the original installation package. It can then be executed normally by the users to install and update their deployed language pack in the deployed location (usually C:\Program Files\Oracle\Explorer Extensions\languages\Updates).

```
explorer_extension_setup.exe /install_langpack {path_to_a_folder_of_xliff_
chm_files}
```

This command does not generate a new installation package. Instead, it copies the modified files directly to its deployed location. It can be used by the user to update the deployed language pack, provided that the user running the installation has access to the folder where the updated files are located (this can be a network location) during the installation time.

## Uninstalling and Downgrading Oracle Beehive Extensions for Explorer (Non-DMS Process)

Use the following methods to uninstall or downgrade Oracle Beehive Extensions for Explorer.

### Uninstalling Oracle Beehive Extensions for Explorer

End users may remove Oracle Beehive Extensions for Explorer from their computers by selecting **Add/Remove Programs** from the Control Panel, selecting Oracle Beehive Extensions for Explorer from the list, and clicking **Remove**.

### Downgrading Oracle Beehive Extensions for Explorer

If you want to install downgraded versions of Oracle Beehive Extensions for Explorer on end users' systems, then simply run the installer older Oracle Beehive Extensions for Explorer version.



---

## Installing Oracle Beehive Conferencing Client

This chapter describes how to install Oracle Beehive Conferencing clients.

There are two clients supported by Beehive Conferencing:

- **Oracle Beehive Java-based (JavaFX) Conferencing Client**

The Oracle Beehive Java-based Conferencing client (also referred to as JavaFX client), by default, is the primary client used to participate in Beehive conferences. However, a user can customize the primary client through Oracle Beehive Central.

When a user joins a conference from a calendar event, e-mail notification, or directly from the Oracle Beehive Conferencing Web Center, the user will be brought to the Oracle Beehive Conferencing Web Center where they will automatically launch into Java-based conferencing application. Being Java-based, the installation of the client is done automatically, with all the necessary Java and JavaFX components. End users are not required to install or deploy any components directly on their machines themselves.

- **Oracle Beehive Conferencing Desktop Client**

The Oracle Beehive Conferencing Desktop Client is a desktop-based application that requires a local installation on each machine. It is only recommended for users using platforms not supported by the JavaFX client.

This section describes how to deploy the JavaFX Conferencing client and the Oracle Beehive Conferencing desktop client using Oracle Beehive's Device Management Service (DMS). This deployment method is recommended for all desktops with standard environments. Currently a desktop based, non-DMS installation method is not available for custom installation.

This chapter covers the following topics:

- [System Requirements](#)
- [Installing JavaFX Conferencing Client](#)
- [Installing Oracle Beehive Conferencing Desktop Client Using Remote Downloader](#)
- [Device Management Service \(DMS\) Based Installation](#)

### System Requirements

This section describes the software and hardware requirements for the JavaFX Conferencing and the Oracle Beehive Conferencing client. For updated list of certified hardware platforms and operating system version, review the certification matrix on the My Oracle Support Web site at the following URL:

<http://support.oracle.com/>

### JavaFX Conferencing Client Operating System

- Microsoft Windows Vista: Home, Business, or Ultimate
- Microsoft Windows XP: Home or Professional
- Microsoft Windows 7 (32-bit and 64-bit edition) - Home Premium, Professional, Ultimate or Enterprise

---

**Note:** Write privileges are required for the destination folder of the installation path of Oracle Beehive Conferencing.

Microsoft Installer framework version 2.0 or above is required for Windows installations. This framework is included in Microsoft Windows XP and later versions of Microsoft Windows.

---

- Apple Mac OS X 10.6 (Snow Leopard) and 10.7 (Lion)

---

**Note:** The Java-based Conferencing Client for the Macintosh only supports Intel processors; PowerPC processors are not supported.

---

- Linux Ubuntu 10.x

### Other Components for JavaFX Conferencing Client

- Java Runtime Environment 1.6.0.23 or later. The end user will be prompted to upgrade if the Java version in their machine does not match the requirements.
- Mac and Linux users may need to set the Web proxy in the Java Control Panel to an explicit proxy.

### Oracle Beehive Conferencing Client Operating System

- Microsoft Windows Vista: Home, Business, or Ultimate
- Microsoft Windows XP: Home or Professional
- Microsoft Windows 7: Home Premium, Professional, Ultimate or Enterprise

---

**Note:** Write privileges are required for the destination folder of the installation path of Oracle Beehive Conferencing.

Microsoft Installer framework version 2.0 or above is required for Windows installations. This framework is included in Microsoft Windows XP and later versions of Microsoft Windows.

---

- Apple Mac OS X 10.5 (Leopard), 10.6 (Snow Leopard), and 10.7 (Lion)

---

**Note:** The Desktop Conferencing Client for the Macintosh is supported on machines with either Intel or PowerPC (minimum 1GHz G4) processors.

---

### Disk Space

- 100 MB minimum

- Hard disk usage varies according to configuration. Custom installation options may require more or less hard disk space.

## Device Management Service (DMS) Based Installation

Oracle recommends that you install and deploy the Oracle Beehive Conferencing client using Oracle Beehive's Device Management Service (DMS). This installation method is recommended for all desktops with standard environments. It is designed for centralized installation and management of auto-updates.

This section covers the following topics:

- [Installing JavaFX Conferencing Client](#)
- [Installing Oracle Beehive Conferencing Desktop Client Using Remote Downloader](#)
- [Updating Oracle Beehive Conferencing in DMS Repository](#)
- [Uninstalling and Downgrading Oracle Beehive Conferencing Client](#)

### Installing JavaFX Conferencing Client

The Java-based architecture JavaFX Conferencing Client can be used by users without requiring any Operating System specific desktop installation. The advantages of this new architecture is the ability to leverage Java to auto-detect, auto-install, and auto-update all the necessary components required by Beehive Conferencing to run. Therefore, for a user to join a conference, the Beehive Conferencing application needs to access:

- JavaScript code (deployJava.js) for launching Java applications and detecting JRE versions

#### Deploying the JavaFX Client

Beehive Conferencing uses the Java resources by accessing the files through java.com site over the Web. This may be a cause for concern if you do not have access to the internet, or cannot access the java.com site (because it is unavailable at that time), and/or do not have proxy settings configured correctly.

To avoid this issue, Oracle recommends that users distribute the resources through the Beehive Desktop Management Service as a DMS package.

1. Configure the Web Conferencing Center to use deployJava.js from DMS:

```
beectl modify_property
--component _BeehiveConferencingService
--name ConfJavaDeployJavaURL
--value DMS://deployJava.js
```

2. Activate the changes:

```
beectl activate_configuration
```

### Installing Oracle Beehive Conferencing Desktop Client Using Remote Downloader

You may install the Oracle Beehive Conferencing Desktop client by distributing a remote downloader to end users. The remote downloader enables end users to download and install the provisioned Oracle Beehive Conferencing client application from the Device Management Service (DMS) repository. By default, the DMS contains a pre-seeded Oracle Beehive Conferencing client application that is provisioned for all end users.

### Obtaining Remote Downloader

Obtain the Oracle Beehive Conferencing client remote downloader, `BeehiveConferencingSetup.exe`, from the directory `<Oracle Beehivehome>/beehive/seed/dm/confclient_win_bootstrapper.zip`.

You may also obtain the remote downloader from Oracle Beehive Central, a Web-based client that provides users a central location to download supported clients and set their preferences for Oracle Beehive functionality.

To download and install Oracle Beehive Conferencing:

1. Access Oracle Beehive Central and click Downloads.
2. On the Oracle Beehive Central Downloads page, click the button next to Oracle Beehive Conferencing. (Whether you use a Windows machine or a Mac, the button will read either "**Windows**" or "**Mac**").

Oracle Beehive Conferencing download page appears displaying product details and installation information.

3. Verify your system requirements, then click **Download for [Windows or Mac]**.

## Updating Oracle Beehive Conferencing in DMS Repository

Follow these steps to update the version of the Oracle Beehive Conferencing client in the DMS repository:

1. Ensure the application package for the updated version of the Oracle Beehive Conferencing client is accessible to the Oracle Beehive server.
2. Upload the updated version of the Oracle Beehive Conferencing client to the DMS repository by running the following command:

```
beectl upload_client_application --file <path to Oracle Beehive Conferencing zip file>
```

---

---

**Note:** This command uploads the new version of the Oracle Beehive Conferencing client application to the DMS repository and makes it available for those who are already provisioned to it.

If you have not provisioned the Oracle Beehive Conferencing client, run the provisioning commands (such as `beectl add_client_application_provisioning`). Refer to "Managing the Device Management Service" in *Oracle Beehive Administrator's Guide* for more information.

---

---

---

---

**Note:** End users may also re-run the remote downloader to install any updated version uploaded to the DMS repository. If the version of the Oracle Beehive Conferencing client in the DMS repository is the same as the one installed in an end user's computer, re-running the remote downloader simply reinstalls the application.

---

---

## Updating Oracle Beehive Conferencing Through DMS

After you update the version of the Oracle Beehive Conferencing client in the DMS repository, when an end user starts the client, Oracle Beehive will automatically check, download, and install any updates from the DMS repository. Oracle Beehive always checks for a newer version when the Oracle Beehive Conferencing client is started.

## Uninstalling and Downgrading Oracle Beehive Conferencing Client

Use the following methods to uninstall or downgrade the Oracle Beehive Conferencing client.

### Uninstalling Oracle Beehive Conferencing Client

The end users may uninstall the Oracle Beehive Conferencing client from their computers by selecting **Add/Remove Programs** from the **Control Panel**, selecting **Oracle Beehive Conferencing** from the list, and clicking **Remove**.

### Downgrading Oracle Beehive Conferencing Client

The end users are always forced to upgrade or downgrade the Oracle Beehive Conferencing client depending on the version in the DMS repository.



---

## Configuring Oracle Beehive Integration for Zimbra

Oracle Beehive Integration for Zimbra is installed and configured in the same Oracle home as Oracle Beehive if you have installed Oracle Beehive on a computer with at least 3 gigabytes (GBs) of memory and you select the Server + Client template during the installation process. If you have installed Oracle Beehive Integration for Zimbra in a different Oracle home than your Oracle Beehive instance, then it 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.

This chapter covers the following topics:

- [Setting Screen Resolution for Oracle Beehive Integration for Zimbra Client](#)
- [Configuring Oracle Beehive Integration for Zimbra for SSL-Enabled Oracle Beehive Deployment](#)
- [Changing HTTP Port for Oracle Beehive Integration for Zimbra](#)

### Setting Screen Resolution for Oracle Beehive Integration for Zimbra Client

Ensure that the screen resolution of all computers running the Oracle Beehive Integration for Zimbra client is 1024 x 768 or higher.

### Configuring Oracle Beehive Integration for Zimbra for SSL-Enabled Oracle Beehive Deployment

If you have configured TLS or SSL for your Oracle Beehive deployment, follow these steps to configure Oracle Beehive Integration for Zimbra (these steps are similar to "[Configuring SSL with Self-Signed 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: Enabling Auto Login Mode for Default Wallet"](#) for directions to create a wallet.

4. If you are using self-signed (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 self-signed (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.

---

## Changing HTTP Port for Oracle Beehive Integration for Zimbra

If Oracle Beehive Integration for Zimbra and your Oracle Beehive instance are on **different** hosts, then use the command `beectl modify_port --protocol HTTP` to change the HTTP port. However, do not use this command if Oracle Beehive Integration for Zimbra and your Oracle Beehive instance are on the same host **and** installed in separate Oracle homes; this will configure all your Oracle Beehive instances to use the same port, which will create a port conflict.

If Oracle Beehive Integration for Zimbra and your Oracle Beehive instance are on the **same** host, change the HTTP port for Oracle Beehive Integration for Zimbra by changing the HTTP port of your Oracle Beehive instance.

---

**Note:** If you want to change the HTTP port number of your Oracle Beehive instance to a value less than 1024, refer to ["Changing HTTP Port"](#) in ["Oracle Beehive Post-Installation Procedures"](#).

---

1. 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.

2. 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 7779:

```

beectl modify_property
 --component ohs_site1.example.com --name HttpListenPort --value 7779

```

3. 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 7779

```

4. 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.

---

## Tuning Oracle Beehive to Improve Performance of Oracle Beehive Integration for Zimbra

Depending on your Oracle Beehive deployment, you may perform the following to improve the performance of Oracle Beehive Integration for Zimbra:

- [Disabling ORMISS](#)
- [Disabling SSL](#)

## Disabling ORMIS

Oracle Remote Method Invocation over Secure Socket Layer (ORMIS) is ORMI over SSL. 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, for security purposes, Oracle Beehive is ORMIS enabled. Consequently, SSL is used for all RMI communication, which adds a significant overhead to your system. ORMIS is required if you configure Oracle Beehive with SSL as described in ["Configuring SSL"](#).

You may not require ORMIS if you have no requirement to encrypt traffic between your application tiers and absolutely no intrusion or eavesdropping risk exists among your application tiers. You may disable ORMIS by following the steps described in the section ["Disable ORMIS"](#) in ["Configuring TLS with Oracle Wallet"](#).

## Disabling SSL

Disabling SSL can improve the performance of Oracle Beehive Integration for Zimbra. If you do not require SSL in your Oracle Beehive deployment, follow these steps to disable SSL:

1. Retrieve the identifier for the component `HttpServerCluster`:

```
beectl list_components --type HttpServerCluster
```

2. Set the value of `HttpServerSslEnabled` in the `HttpServerCluster` component to `false`, then run `beectl modify_local_configuration_files`:

```
beectl modify_property
--component <HttpServerCluster identifier retrieved from previous step>
--name HttpServerSslEnabled
--value false
```

3. Activate the configuration and commit changes:

```
beectl activate_configuration
beectl modify_local_configuration_files
```

---

# Cloning Oracle Beehive Application Tiers and Sites

This chapter covers the following topics:

- [Introduction to Cloning](#)
- [Preparing Source Application Tier Instance](#)
- [Application Tier Cloning](#)
- [Site Cloning](#)
- [OC4J Instance Cloning](#)
- [Oracle Beehive Integration for Zimbra Cloning](#)

This chapter 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/PS 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](#)".

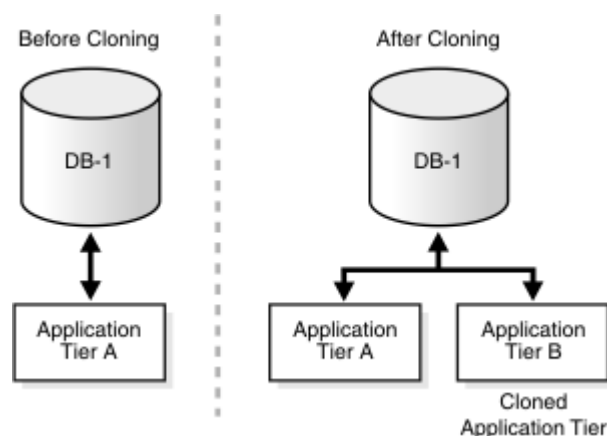
### Application Tier Cloning

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_mdtier` 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 29–1 Application Tier Cloning**



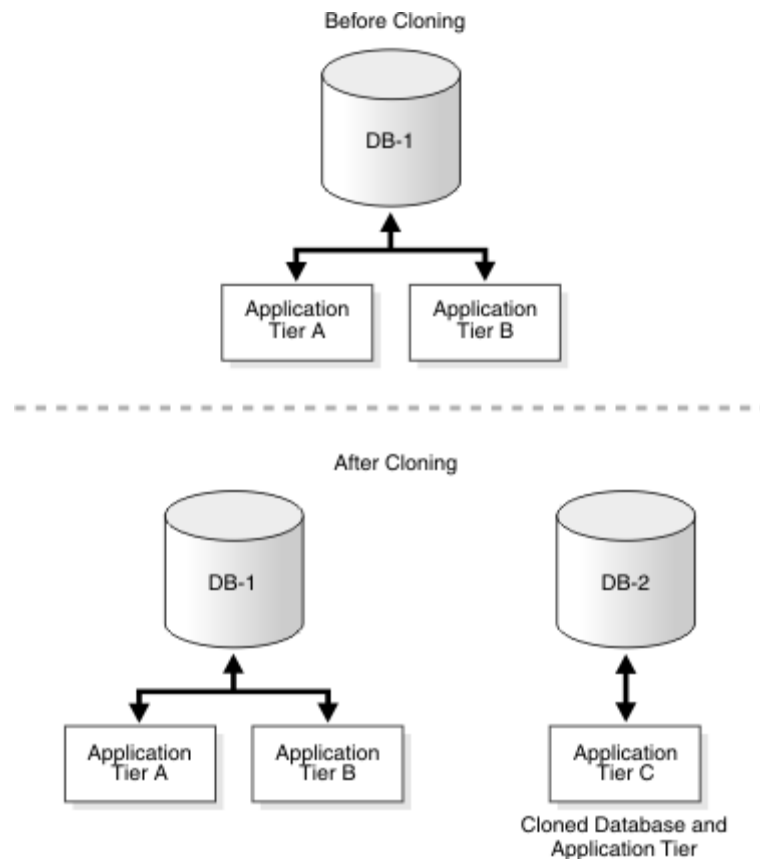
## Site Cloning

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 29-2 Site Cloning**



---

**Note:** As [Figure 29–2, "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. For more information, refer to ["Cloned Application Tiers Are Not Automatically SSL or AJPS Enabled"](#).

---

### **OC4J Instance Cloning**

OC4J instance cloning enables 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/Activate Any Pending Configuration Changes to Central Configuration Repository](#)
- [Step 3: Shut Down All Processes On the Application Tier](#)
- [Step 4: Call beectl clone\\_preparation Command](#)
- [Step 5: Zip Files to Create Clone Image](#)

### **Step 1: Verify Requirements**

Ensure that the chipset and the operating system version (including any operating system patches) of the source computer is the same as the destination computer. (The source computer 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/Activate Any Pending Configuration Changes to 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: 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 4: 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 29–1** *beectl clone\_preparation Options*

Option	Mandatory/ Optional	Description
<code>--file</code>	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><b>Note:</b> This text file <b>must not</b> be located in the Oracle home because you may receive warnings or errors from some zip tools about zipping an open file.</p>

### Step 5: 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. Use a file archiver tool that can archive and compress a list of files that total at least 2 GB in size. Also, make sure that the tool preserves the permissions and timestamps of the files. For example, you may use the tool 7-Zip, which you may download from <http://www.7-zip.org/>.

Use the following command to archive the Oracle Beehive home with the 7-Zip tool:

```
C:\7-zip\7z.exe a C:\clone_beehive.7z @C:\clone_prepare.txt
```

`C:\clone_prepare.txt` is the file generated by the `beectl clone_preparation` command. `C:\clone_beehive.7z` is the name of the file that contains the archived Oracle Beehive home.

---

---

**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.

---

---

Alternatively, you may use the `tar` and `gzip` commands.

The following example shows how to archive and compress the source on Linux:

```
cd <source Oracle home>
tar -c -T <file created by beectl clone_preparation command> -f - | gzip >
clone_image.tar.gz
```



---

**Note:** In some Solaris environments, because of `stdout` limitations, the `gzip` command may not be able to create a tar file of the Oracle home properly. In this case, use two commands to archive and compress the Oracle home:

1. Create a tar file:

```
cd $ORACLE_HOME
tar cf /tmp/clone_beehive.tar -I /tmp/clone_prepare.txt
```

2. Compress the tar file with `gzip`:

```
gzip /tmp/clone_beehive.tar
```

The `tar` command may issue warnings if the sticky bit is set on some files. You can safely ignore these warnings.

Ensure that the file created by the `tar` command (in the previous example, this would be `clone_image.tar.gz`) is not located in the Oracle home; the `tar` command may fail.

You may receive an error message similar to the following when using this `tar` command:

```
tar: sqljdbc.dll: Cannot stat: No such file or directory
tar: instjdbc.sql: Cannot stat: No such file or directory
tar: sqljdbc.dll: Cannot stat: No such file or directory
tar: instjdbc.sql: Cannot stat: No such file or directory
tar: pool.jar: Cannot stat: No such file or directory
tar: Error exit delayed from previous errors
```

If you encounter this error, shut down all processes on your Oracle Beehive application tier with the `beectl stop --all` command before creating the clone image again.

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

---

**WARNING:** Ensure that you have created a compressed Oracle home by following the steps described in ["Preparing Source Application Tier Instance"](#).

Do not simply recursively copy the Oracle home from the source application tier to the destination computer. The steps described in ["Preparing Source Application Tier Instance"](#) ensure that all required files have been copied properly. (Depending on your operating system, the recursive copying operation may not be able to copy certain files.)

---

1. Copy the compressed Oracle home from the source computer to the destination computer.
2. Extract the compressed Oracle home into a directory, which will become the new Oracle home at the destination location.

If you are using 7-Zip as your file archiver tool, extract the compressed Oracle home with the following commands:

```
cd C:\new_oracle_home
C:\7-zip\7z.exe x -r C:\clone_beehive.7z
```

If you are using tar and gunzip, extract the compressed Oracle home with the following commands:

```
mkdir -p <destination Oracle home>
cd <destination Oracle home>
gunzip < <directory containing tar file>/clone_image.tar.gz | tar xf -
```

---

**Note:** Ensure that the tar and gzip/gunzip versions on the source and destination computers 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 Linux, see the file /etc/oraInst.loc. For Solaris, 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 5.8.3 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:

```
export PERL5LIB=$ORACLE_HOME/perl/lib/5.8.3/i686-linux-thread-multi:
$ORACLE_HOME/perl/lib/5.8.3:
$ORACLE_HOME/perl/lib/site_perl/5.8.3/i686-linux-thread-multi/
```

## Step 3: Modify Oracle Home Path

---

**Note:** This step applies only to UNIX-based operating systems.

---

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 29–2** *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

1. Ensure that no executables are running on the target application tier. For example, if `<target application tier Oracle home>/bin/sqlplus` is running, then the cloning process will fail.

To verify this, run the command `ps -auxwww | grep -i $ORACLE_HOME`, where `$ORACLE_HOME` is the Oracle home of the target application tier.

2. Run the `beectl clone_midtier` command as described in the following section.

### **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 29–3** *beectl clone\_midtier Options*

Option	Mandatory/Optional	Description
<code>--ias_instance_name</code>	Mandatory	<p>The instance name for the clone.</p> <p><b>Notes:</b> 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 computer 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><b>Note:</b> 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>

**Table 29–3 (Cont.) beectl clone\_midtier Options**

Option	Mandatory/ Optional	Description
--oui_inv_ptr_loc	Optional  <b>Note:</b> This option is available only for UNIX-based systems.  Do not specify if it does not exist on the computer 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.  <b>Note:</b> This option is available only for UNIX-based systems.  For more information, refer to " <a href="#">Oracle Inventory Location Option of Clone Commands on UNIX-Based Systems</a> ".  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:  --oui_inv_ptr_loc "/etc/oraInst.loc"  <b>Note:</b> This value is platform-dependent. On Linux, it is /etc/oraInst.loc.
--oracle_home_name	Optional	Oracle home name. The default value is the value for the --ias_instance_name option.
--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 beectl clone_site for the first time from a source site that has an external source enabled, you will receive a warning message similar to the following:  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  For this example, you would disable your directory profile before continuing.

**Table 29–3 (Cont.) beectl clone\_midtier Options**

Option	Mandatory/ Optional	Description
--site_key	Mandatory	<p>Site key of the Oracle Beehive application tier you are cloning. This is an alphanumeric string.</p> <p><b>Note:</b> If you are not in shell mode, you must obfuscate the site key and add the --obfuscated option to the beectl clone_midtier command.</p> <p>To obfuscate a password, use the beectl obfuscate command:</p> <pre>beectl obfuscate   --expiration_time_in_minutes 0 Enter value for password:</pre> <p>Successfully obfuscated the string.</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.

---

1. Ensure that no executables are running on the target application tier. For example, if `<target application tier Oracle home>/bin/sqlplus` is running, then the cloning process will fail.

To verify this, run the command `ps -auxwww | grep -i $ORACLE_HOME`, where `$ORACLE_HOME` is the Oracle home of the target application tier.

2. Run the `beectl clone_site` command as described in the following section.

### `beectl clone_site` Command

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 property `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.
  - It deletes the configuration of `UnmanagedOc4j` from the central configuration repository. In particular, Oracle Beekeeper and its configuration are deleted.
- 

The following table describes the options of the `beectl clone_site` command:

**Table 29–4** *beectl clone\_site Options*

Option	Mandatory/ Optional	Description
--ias_instance_name	Mandatory	<p>The instance name for the clone.</p> <p><b>Notes:</b> 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>
--host_name	Mandatory	<p>The hostname of the computer on which the clone is being created. This must be the fully qualified hostname (with the domain appended). For example, hostB.example.com.</p>
--db_connect_string	Mandatory	<p>Database connect string for the new site. This would be the connect string for the cloned database.</p> <p><b>Note:</b> The database connect string cannot contain any new line characters (\n, \r, or \n\r); the connect string must be a single-line value.</p>
--db_schema_password	Mandatory	<p>Database password for the schema.</p> <p><b>Note:</b> If you are not in shell mode, you must obfuscate the database password and add the --obfuscated option to the beectl clone_site command.</p> <p>To obfuscate a password, use the beectl obfuscate command:</p> <pre>beectl obfuscate   --expiration_time_in_minutes 0 Enter value for password:</pre> <p>Successfully obfuscated the string.</p>
--db_schema_name	Optional	<p>New database schema name. Typically, this would be the same schema as the old site, which is usually BEE_CODE.</p>



Table 29–4 (Cont.) *beectl clone\_site* Options

Option	Mandatory/ Optional	Description
--db_rac_node_information	Optional	<p>New values for the host:port of Oracle RAC nodes.</p> <p>The host name should <b>not</b> be the VIP hostname. Specify the actual computer 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>&lt;Oracle RAC database home&gt;/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 an Oracle 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>
--retain_rac_node_information	Optional	Retain existing values for db_rac_node_information. This option cannot be specified with --db_rac_node_information.
--oracle_home_name	Optional	Oracle home name. The default value is the value provided for the --ias_instance_name option.
--oui_inv_ptr_loc	Optional. <b>Note:</b> This option is available only for UNIX-based systems.  Do not specify if it does not exist on the computer from which you are running this command; in this case, the Oracle inventory will be created in the user's home directory.	<p>Oracle Universal Installer inventory location.</p> <p><b>Note:</b> This option is available only for UNIX-based systems.</p> <p>For more information, refer to <a href="#">"Oracle Inventory Location Option of Clone Commands on UNIX-Based Systems"</a>.</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 "/etc/oraInst.loc"</pre> <p><b>Note:</b> This value is platform-dependent. On Linux, it is <code>/etc/oraInst.loc</code>.</p>

**Table 29–4 (Cont.) beectl clone\_site 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 beectl clone_site 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>
--site_name	Optional	<p>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.</p>
--ignore_validation_warnings	Optional	<p>If you specify this option, the site cloning process will proceed regardless of warnings about target application tiers referring to external resources. Refer to <a href="#">"Step 5: Prevent Services from Target Application Tiers from Referring to External Resources"</a> for more information about external resources.</p>
--site_key	Mandatory	<p>Site key of the site you are cloning. This is an alphanumeric string.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>■ The site key you specify must be the same as the source site.</li> <li>■ If you are not in shell mode, you must obfuscate the site key and add the --obfuscated option to the beectl clone_midtier command.</li> </ul> <p>To obfuscate a password, use the beectl obfuscate command:</p> <pre>beectl obfuscate --expiration_time_in_minutes 0 Enter value for password:</pre> <p>Successfully obfuscated the string.</p>

## 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](#)

---

**Note:** 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 an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide*), 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 the section "Configuring Authentication Service to Use LDAP Server" in "Integrating an External User Directory with Oracle

Beehive" in *Oracle Beehive Integration Guide*, 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

Perform the steps described in "[Step 5: Perform Miscellaneous Operations](#)".

Afterwards, ensure that the cloned database has the same name as the original database name. This is required to ensure that the Change Data Capture works properly. If the cloned database name is different from the original database name, change GLOBAL\_NAME with the command `ALTER DATABASE RENAME GLOBAL_NAME`. Refer to *Oracle Database SQL Language Reference* for more information.

## OC4J Instance Cloning

OC4J instance cloning enables 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 29–5** *beectl clone\_oc4j\_instance Options*

Option	Mandatory/ Optional	Description
--source_oc4j_instance_id	Either this option or --source_oc4j_instance_name is required	ID of the managed component to be cloned, for example, BEEAPP_instance1.example.com
--source_oc4j_instance_name	Either this option or --source_oc4j_instance_id is required	Prefix of the managed component to be cloned, for example BEEAPP, oc4j_soa, or BEECORE
--target_oc4j_instance_name	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 BEEAPP_CLONE, its ID will be similar to BEEAPP_CLONE_instance1.example.com.
--exclusion_list	Optional	List of services to exclude from the newly created OC4J instance
--working_list	Optional	List of services that will only be deployed on the newly created OC4J instance.

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 except for ClientManagementService:

```
beectl clone_oc4j_instance
 --source_oc4j_instance_name BEEAPP
 --target_oc4j_instance_name BEEAPP_SOURCE
 --exclusion_list ClientManagementService
```

## Oracle Beehive Integration for Zimbra Cloning

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_middtier` 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_middtier` 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 5: 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_middtier` 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 Linux, the Oracle inventory location pointer file is `/etc/oraInst.loc`:

```
prompt>cat /etc/oraInst.loc
inventory_loc=/private/beehive/oraInventory
inst_group=g900
```

- The value for `inventory_loc` must be a valid existing location, or the parent directory of this value must exist. In this example, `/private/beehive` must exist.

- Any user installing Oracle software must belong to the operating system group specified by `inst_group`.

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 (`/etc/oraInst.loc` on Linux) 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 (`/etc/oraInst.loc` on Linux), 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 Transport Layer Security (TLS), 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](#)

## Cloning SSL-Enabled Application Tiers

If you have cloned an SSL-enabled application tier with self-signed certificates, then follow these steps to enable SSL for your cloned application tier:

1. Recreate the self-signed certificates on the cloned application tier.
2. Perform in "[Configuring TLS with Oracle Wallet](#)".

If you have cloned an SSL-enabled application tier with test certificates, you only need to perform "[Step 2: Configuring Oracle Beehive Instance to Use Oracle Wallet](#)" in "[Configuring TLS with Oracle Wallet](#)".

## 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 an External User Directory with Oracle Beehive](#)" in *Oracle Beehive Integration Guide*), 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 the section "Active Directory Considerations" in "Integrating an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide*, 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 *Oracle Application Server Installation Guide for Linux x86*.
  - 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 Directory, 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
-----+-----
BEEGMGT_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 BEEGMGT_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 change log 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 an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide*.

## 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 AJPS Enabled](#)" for more information.

---

---

**Note:** You cannot clone any Oracle Beehive DMZ or Oracle Beekeeper instances. You must reinstall any Oracle Beehive DMZ or Oracle Beekeeper instance in your cloned site.

---

---

## 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 Beekeeper Post-Installation Procedures

---

Depending on your security requirements or any other issue particular to your deployment, perform one or more of the following procedures after installing Oracle Beekeeper:

- [Configuring Oracle Beekeeper for SSL Access](#)
- [Configuring Oracle Beekeeper with Oracle Wallet](#)
- [Changing Oracle Beekeeper Port](#)
- [Extending Java Single Sign-On Session Timeout](#)
- [Configuring Oracle Beekeeper for LDAP-Based Authentication](#)
- [Configuring Virtual Host](#)
- [Restarting OC4J](#)

### Configuring Oracle Beekeeper for SSL Access

1. Create a keystore with the `keytool` command. This command is located in `<Oracle Beekeeper home>/jdk/bin`. The following example will create a keystore named `server.keystore` in the directory `<Oracle Beekeeper home>` with the password `welcome`:

```
keytool -genkey -keyalg "RSA"
-keystore <Oracle Beekeeper home>/server.keystore
-storepass welcome -validity 90
```

---

**Note:** For more information about the `keytool` utility, refer to <http://download.oracle.com/javase/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. In `<Oracle Beekeeper home>/j2ee/home/config/default-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.

---

3. Add the following element inside the `<web-site>` element:

```
<ssl-config
 keystore="<Oracle Beekeeper 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.

4. Restart OC4J as described in ["Restarting OC4J"](#).

## Configuring Oracle Beekeeper with Oracle Wallet

If you have configured Oracle Beehive for SSL access and you want to add or modify a directory profile, you must specify the location of an Oracle Wallet configured for Oracle Beehive in the `UnmanagedOc4j` component in which Oracle Beekeeper runs.

---

**Note:** For information about directory profiles (or LDAP mapping profiles), refer to "Step 1: Creating an LDAP Mapping Profile" in "Integrating an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide*.

For more information about configuring Oracle Wallet for Oracle Beehive, refer to ["Step 2: Configuring Oracle Beehive Instance to Use Oracle Wallet"](#) in ["Configuring TLS with Oracle Wallet"](#).

---

1. Retrieve the component identifier for `UnmanagedOc4j` with the `beectl list_components` command. You may call this command from any Oracle Beehive application tier:

```
beectl list_components --type UnmanagedOc4j
```

```
-----+-----
Component type | Component identifier
-----+-----
UnmanagedOc4j | e084c8c8-4a78-4852-8699-480b9bf4f79c
-----+-----
```

2. Set the `WalletDir` property in the `UnmanagedOc4j` component to the location of your Oracle Wallet:

```
beectl modify_property
 --component <UnmanagedOc4j identifier>
 --name WalletDir
 --value <Oracle Wallet directory>
```

3. Activate the configuration:

```
beectl activate_configuration
```

## Changing Oracle Beekeeper Port

1. Edit the file `<Oracle Beekeeper 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"](#).

## Changing Oracle Beekeeper RMI Port

1. Edit the file `<Oracle Beekeeper home>/j2ee/home/config/rmi.xml` and modify the port attribute in the `<rmi-server>` element to any available port number:

```
<rmi-server ... port="23792" ... >
```

2. Restart OC4J as described in ["Restarting OC4J"](#).

3. Retrieve the component identifier for UnmanagedOc4j with the `beectl list_components` command. You may call this command from any Oracle Beehive application tier:

```
beectl list_components --type UnmanagedOc4j
```

```
-----+-----
Component type | Component identifier
-----+-----
UnmanagedOc4j | e084c8c8-4a78-4852-8699-480b9bf4f79c
-----+-----
```

4. Set the RmiPort property in the UnmanagedOc4j component to the same port number you specified in `rmi.xml`:

```
beectl modify_property
--component <UnmanagedOc4J identifier>
--name RmiPort
--value 23792
```

5. Activate the configuration:

```
beectl activate_configuration
```

## Extending Java Single Sign-On Session Timeout

To extend the Java Single Sign-On session timeout for Oracle Beekeeper, add the highlighted text to the file `<Oracle Beekeeper home>/j2ee/home/config/jps-config.xml`:

`home>/j2ee/home/config/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.jssso.SSOCookieTokenAssertter"/>
 <property name="idm.token.collector.class"
 value="oracle.security.jps.internal.jssso.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 Beekeeper for LDAP-Based Authentication

You may configure Oracle Beekeeper so that it authenticates its users with credentials stored in your LDAP directory. Refer to "Integrating an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide* for more information.

---

**Note:** If you set a directory profile as default in Oracle Beekeeper, you must configure Oracle Beekeeper for LDAP-based authentication.

---

1. Edit the file `<Oracle Beekeeper 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.

Refer to the section "Retrieving Information About the LDAP Server" in "Integrating an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide* for more information about these properties.

```
<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="orclUserV2"/>
 <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="orclGroup"/>
 <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:** The property

oracle.security.jaas.ldap.user.object.class corresponds to the UserObjectClass property. The property

oracle.security.jaas.ldap.role.object.class corresponds to the GroupObjectClass property. Refer to the section "Default UserObjectClass and GroupObjectClass Values" in "Integrating an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide* for more information.

Follow these steps to obtain values for jps-config.xml:

1. Run the command `beectl modify_local_configuration_files`.
2. Obtain the required values from the following file:

```

<Oracle Beehive home>/j2ee/
 <any OC4J container>/config/system-jazn-data.xml

```

For example,

```

<Oracle Beehive home>/j2ee/BEEAPP/config/system-jazn-data.xml

```

Look for the <application> element that contains the element <name>beehive-auth-framework-password</name>.

Note that you must run the command `beectl modify_local_configuration_files` before obtaining values from `system-jazn-data.xml`; the LDAP-related entries will not appear in this file until you do so.

2. In the same file (<Oracle Beekeeper home>/j2ee/home/application-deployments/**javasso**/jps-config.xml), update the <jpsContexts> element with the highlighted value:

```

<jpsContexts default="ldap">

```

3. Edit the file (<Oracle Beekeeper home>/j2ee/home/application-deployments/**beekeeper**/jps-config.xml) and make the same changes as in step 1.

4. In the same file (<Oracle Beekeeper home>/j2ee/home/application-deployments/**beekeeper**/jps-config.xml), update the <jpsContexts> element with the highlighted value:

```

<jpsContexts default="jssso-ldap">

```

5. Modify the AuthStoreType property of the Authentication Service to ldap with the `beectl modify_property` command. Refer to the section "Configuring Authentication Service to Use LDAP Server" in "Integrating an External User Directory with Oracle Beehive" in *Oracle Beehive Integration Guide* for more information.

---

**Note:** You may also modify this property through Oracle Beekeeper.

---

6. Activate the configuration and commit changes:

```
beectl activate_configuration
beectl modify_local_configuration_files
```

---

**Note:** Although you may activate the configuration through Oracle Beehive, you must run the command `beectl modify_local_configuration_files` from the command line.

---

7. Edit the file `<Oracle Beekeeper home>/j2ee/home/application-deployments/javasso/jps-config.xml` and update the value of `oracle.security.jaas.ldap.provider.credential` by copying the obfuscated password from `<Oracle Beehive home>/j2ee/<any OC4J container>/config/system-jazn-data.xml`. The obfuscated password is in the `<application>` element that contains the element `<name>beehive-auth-framework-password</name>`.
8. Restart OC4J as described in ["Restarting OC4J"](#).

## Reconfiguring Oracle Beekeeper for UDS-Based Authentication

Follow these steps to reconfigure Oracle Beekeeper to use User Directory Services (UDS) instead of LDAP for authentication.

---

**Note:** If you set a directory profile as non-default in Oracle Beekeeper, configure Oracle Beekeeper for UDS-based authentication.

---

1. In the file `<Oracle Beekeeper home>/j2ee/home/application-deployments/javasso/jps-config.xml`, update the `<jpsContexts>` element with the highlighted value:

```
<jpsContexts default="db">
```

2. In the file `<Oracle Beekeeper home>/j2ee/home/application-deployments/beekeeper/jps-config.xml`, update the `<jpsContexts>` element with the highlighted value:

```
<jpsContexts default="jssso-db">
```

3. Modify the `AuthStoreType` property of the Authentication Service to `db` 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 db
```

4. Activate the configuration and commit changes:

```
beectl activate_configuration
beectl modify_local_configuration_files
```

---

**Note:** You may activate the configuration through Oracle Beekeeper. However, you must use the `beectl` command to modify local configuration files.

---

5. Restart OC4J as described in ["Restarting OC4J"](#).

## Configuring Virtual Host

You may configure multiple instances of Oracle Beekeeper with a virtual host through a load balancer so that all your Oracle Beekeeper instances will be accessed by a single point of access, the virtual host through a load balancer.

---

**Note:** If you have multiple Oracle Beekeeper nodes behind your load balancer's virtual host, then you must configure the load balancer virtual host so that it uses cookie insert persistence in order to maintain server affinity.

Refer to the documentation of your load balancer for more information about cookie insert persistence.

---

1. Edit the file `<Oracle Beekeeper home>/j2ee/home/config/default-web-site.xml` and specify the host name and port number of your load balancer in the `<frontend>` child element of `<web-site>` as follows:

```
<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">
 <default-web-app application="default" name="defaultWebApp" />
 ...
 <frontend host="beehiveadmin.us.oracle.com" port="80" />
</web-site>
```

In this example, `beehiveadmin.us.oracle.com` is the host name of the load balancer and 80 is the port number.

2. If you or an administrator uses a Microsoft Windows computer to access Oracle Beekeeper, then on that computer, edit the file `C:\WINDOWS\system32\drivers\etc\hosts` file and map the IP address of the computer that runs Oracle Beehive with the host name of your virtual host.

For example, if 10.229.175.134 is the host name of the computer that runs Oracle Beekeeper and `beehiveadmin.us.oracle.com` is the host name of your load balancer, then add the following line to your hosts file:

10.229.175.134      beehiveadmin.us.oracle.com

3. Restart OC4J as described in ["Restarting OC4J"](#).

## Restarting OC4J

Restart the Oracle Beekeeper unmanaged OC4J instance with the following commands:

```
<Oracle Beekeeper home>/opmn/bin/opmnctl stopall
<Oracle Beekeeper home>/opmn/bin/opmnctl startall
```

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 IV

---

## Oracle Beehive High Availability Configuration

The Part IV of the Installation Guide describes how to install and configure Oracle Beehive in a high availability environment. It contains the following chapters:

- [Configuring and Installing Oracle Beehive for Oracle RAC](#)
- [Installing Oracle Beehive in High Availability Environment](#)
- [Enabling AJPS](#)



---

## Enabling AJPS

This chapter 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 chapter 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://download.oracle.com/javase/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 -validity <number of days> -storepass 123456
```

In this utility:

- The `keystore` option sets the filename where the keys are stored.
- The `-validity <number of days>` option sets the expiry date.
- 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 `/home/user/cert.txt`:

```
<Oracle home>/jdk/bin/keytool -export -file /home/user/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>
```



For example, if the identifier of your Oracle Beehive instance is `beehive_instance_instance1.example.com` and the full path name of your keystore file is `/home/user/mykeystore.jks`, then run the following command:

```
beectl modify_property
--component beehive_instance_instance1.example.com
--name KeystoreFile
--value /home/user/mykeystore.jks
```

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 and Installing Oracle Beehive for Oracle RAC

This chapter describes how to install and configure Oracle Beehive to support deployment on multiple Oracle Real Application Clusters (Oracle RAC) nodes. It covers the following topics:

- [Naming Conventions](#)
- [Pre-Install Steps](#)
- [Installation](#)
- [Post-Install Steps](#)
- [Installing Multiple Oracle Beehive Instances with Oracle RAC](#)
- [Enabling Oracle Beehive Affinity Service](#)
- [Troubleshooting](#)

---

**Note:** You may use Database Configuration Assistant (DBCA) to create Oracle RAC nodes. However, refer to My Oracle Support 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

If your Oracle RAC database uses raw storage, you must manually create the tablespace before running the Oracle Beehive Install Wizard.

## Installation

---

---

**Note:** If you want to install Oracle Beehive with Oracle Beehive Provisioning Application and you require SSL Oracle Notification Services (ONS) notification, you must follow the steps described in ["Enabling Secure ONS Notification"](#).

For more information about Oracle Beehive Provisioning Application, refer to ["Provisioning Oracle Beehive"](#).

---

---

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:** Use the global service name (MYDB).

---

---

**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. [Update Database System Object](#)
3. [Commit Configuration Changes](#)
4. [Restart Oracle Beehive](#)
5. [Enable Secure ONS Notification](#)

### Step 1 Backup Bootstrap File

The file `<Oracle home>/beehive/conf/beehiveconfig.xml` contains the bootstrap database connect information. Take a backup of this file before applying any post-install steps.

## Step 2 Update Database System Object

Modify the database system object's configuration to specify the ONS remote ports that are listening on your Oracle RAC nodes. The following example specifies two ONS remote ports with the `ons_entry` option:

```
beectl modify_database
 --database _CURRENT_SITE:Database
 --ons_entry ONS_HOST1:ONS_PORT1
 --ons_entry ONS_HOST2:ONS_PORT2
```

To determine the ONS remote port, run the following command:

```
<Oracle Cluster Ready Services home>/bin/onsctl ping
```

You should see output similar to the following. The ONS remote port is indicated by the highlighted text:

```
Number of onsconfiguration retrieved, numcfg = 2
onscfg[0]
 {node = strasha05.us.oracle.com, port = 6200}
Adding remote host strasha05.us.oracle.com:6200
onscfg[1]
 {node = strasha06.us.oracle.com, port = 6200}
Adding remote host strasha06.us.oracle.com:6200
ons is running ...
```

## Step 3 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 4 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_files`) if you modify the ONS details in the database system object's configuration (as you have done previously).

---

---

## Step 5 Enable Secure ONS Notification

If you have not already done so and you require SSL ONS notification, follow the steps described in ["Enabling Secure ONS Notification"](#).

## Enabling 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 ["Provisioning Oracle Beehive"](#) for more information.

---

---

This step involves specifying the Oracle Wallet directory in the `ons.config` file. 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_WALLET_DIR
```

*MY\_SECURE\_WALLET\_DIR* is your Oracle Wallet directory. Refer to ["Configuring TLS with Oracle Wallet"](#) for information about configuring Oracle Wallet.

## 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, Install 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.
2. For one instance, apply all the post-installation steps.
3. For all the other instances, run the following post-installation steps:
  - [Step 1, "Backup Bootstrap File"](#)
  - [Step 3, "Commit Configuration Changes"](#), except run **only** the `beectl modify_local_configuration_files` command.

### Install One Oracle Beehive Instance, Apply Post-Install Steps, Install 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 install additional Oracle Beehive instances.

## Enabling Oracle Beehive Affinity Service

Oracle Beehive Affinity Service enables other Oracle Beehive services to take advantage of the enhanced performance when instances or nodes from your Oracle RAC environment has affinity for any data. Currently, only the event framework, or

the Object Event Framework (OEF), and the E-mail Service can take advantage of the Affinity Service.

To enable Oracle Beehive Affinity Service, follow these steps:

1. Create one database service for each Oracle RAC node with the following command:

```
srvctl add service
-d <database global service name>
-s <database service name>
-r <Oracle RAC node instance ID>
```

---

**Note:** These services should run on only one Oracle RAC node and should not failover to other nodes.

---

For example, suppose your database has the global service name BEEHIVE, and has four Oracle RAC nodes with IDs BEEHIVE1, BEEHIVE2, BEEHIVE3, and BEEHIVE4. Create four database services (named afserv1, afserv2, afserv3, and afserv4) with the following commands:

```
srvctl add service -d BEEHIVE -s afserv1 -r BEEHIVE1
srvctl add service -d BEEHIVE -s afserv2 -r BEEHIVE2
srvctl add service -d BEEHIVE -s afserv3 -r BEEHIVE3
srvctl add service -d BEEHIVE -s afserv4 -r BEEHIVE4
```

2. Start the database services you just created. For example:

```
srvctl start service -d BEEHIVE -s afserv1
srvctl start service -d BEEHIVE -s afserv2
srvctl start service -d BEEHIVE -s afserv3
srvctl start service -d BEEHIVE -s afserv4
```

3. Set the property AffinityServiceNames with the connection descriptors of your newly created database services. For example, to set this property with the four database services created in the previous step, call the following command. The example assumes the domain name is example.com.

---

**Note:** (Line breaks have been inserted in the following example for better readability.)

A connect descriptor cannot contain any new line characters (\n, \r, or \n\r); this connect string must be a single-line value.

---

```
beectl modify_property
--component _CURRENT_SITE:Database
--name AffinityServiceNames
--value "(DESCRIPTION=
 (ADDRESS_LIST=
 (ADDRESS=(PROTOCOL=TCP) (HOST=host1.example.com) (PORT=1521)))
 (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=afserv1.example.com)))"
--value "(DESCRIPTION=
 (ADDRESS_LIST=
 (ADDRESS=(PROTOCOL=TCP) (HOST=host2.example.com) (PORT=1521)))
 (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=afserv2.example.com)))"
--value "(DESCRIPTION=
 (ADDRESS_LIST=
 (ADDRESS=(PROTOCOL=TCP) (HOST=host3.example.com) (PORT=1521)))
```

```
(CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=afserv3.example.com))) "
--value "(DESCRIPTION=
 (ADDRESS_LIST=
 (ADDRESS=(PROTOCOL=TCP) (HOST=host4.example.com) (PORT=1521)))
 (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=afserv4.example.com))) "
```

Refer to ["Modifying AffinityServiceNames Property"](#) for more information about setting this property.

4. Activate the configuration with the following command:

```
beectl activate_configuration
```

The database framework will immediately start using the Affinity Service.

---

**Note:** You must disable SSL ONS notifications. Follow these steps to do so:

1. Set the property NotificationServerSslEnabled to false:

```
beectl modify_property
--component _CURRENT_SITE:OpmnCluster
--name NotificationServerSslEnabled
--value false
```

2. Activate the configuration:

```
beectl activate_configuration
```

To restart a Oracle RAC node with which you have configured Oracle Beehive Affinity Service, follow these steps:

1. Stop the database service associated with the Oracle RAC node you want to restart.
  2. Restart the Oracle RAC node.
  3. Start the associated database service.
- 

## Troubleshooting

### Automated Failover, beectl, and Oracle Beekeeper

If your Oracle RAC database is in the process of performing an automated failover, beectl commands and Oracle Beekeeper instances that are currently running may appear to hang. If this persists for a long period of time, for beectl, terminate the beectl process (with the kill command, for example) and then run the beectl command again later. For Oracle Beekeeper, restart the Oracle Beekeeper unmanaged OC4J instance with the following commands:

```
<Oracle Beekeeper home>/opmn/bin/opmnctl stopall
<Oracle Beekeeper home>/opmn/bin/opmnctl startall
```

### Modifying AffinityServiceNames Property

If the global service name has been created with the domain name (such as example.com), all the database services should use the same domain name even though they have not been created with the domain name.

Alternatively, you may specify the service name instead of the connection descriptor. For example:

```
beectl modify_property
--component _CURRENT_SITE:Database
--name AffinityServiceNames
```



```
--value afserv1.example.com
--value afserv2.example.com
--value afserv3.example.com
--value afserv4.example.com
```

If you use service names instead of connection descriptors, Oracle Beehive will use the connection string specified by the property `ConnectDescriptor` in the `_CURRENT_SITE:Database` component. For example, you would ensure that the `ConnectDescriptor` would be set as follows:

```
beectl list_properties --component _CURRENT_SITE:Database
--format xml
```

```
<?xml version="1.1" encoding="UTF-8"?>
<beectl-output resultset="table">
 <row>
 <column name="Property name">AccessSchemas</column>
 <column name="Property value"></column>
 </row>
 ...
 <row>
 <column name="Property name">ConnectDescriptor</column>
 <column name="Property value">
 (DESCRIPTION=
 (ADDRESS_LIST=
 (LOAD_BALANCE=yes)
 (FAILOVER=on)
 (ADDRESS= (PROTOCOL=TCP) (HOST=host1.example.com)
 (PORT=1521))
 (ADDRESS= (PROTOCOL=TCP) (HOST=host2.example.com)
 (PORT=1521))
 (ADDRESS= (PROTOCOL=TCP) (HOST=host3.example.com)
 (PORT=1521))
 (ADDRESS= (PROTOCOL=TCP) (HOST=host4.example.com)
 (PORT=1521))
)
 (CONNECT_DATA=
 (SERVER=DEDICATED)
 (SERVICE_NAME=beehive.us.oracle.com)))
 </column>
 </row>
```

Oracle Beehive does not support multiple connect strings for the same affinity service. However, Oracle Beehive supports multiple listeners for the same affinity service. For example, suppose you have the following entries:

- **Primary service:**

```
(DESCRIPTION=
 (ADDRESS_LIST= (ADDRESS=
 (PROTOCOL=TCP)
 (HOST=host11-vip.example.com)
 (PORT=32044)))
 (CONNECT_DATA=
 (SERVER=DEDICATED)
 (SERVICE_NAME=beehive.us.oracle.com)))
```

- **Stand-by service:**

```
(DESCRIPTION=
 (ADDRESS_LIST= (ADDRESS=
 (PROTOCOL=TCP)
 (HOST=host2-vip.us.oracle.com)
```

```
(PORT=32044))
(CONNECT_DATA=
 (SERVER=DEDICATED)
 (SERVICE_NAME=beehive.us.oracle.com))
```

Use the following connect string for these two services:

```
(DESCRIPTION=
 (ADDRESS_LIST=
 (ADDRESS=
 (PROTOCOL=TCP)
 (HOST=host1-vip.us.oracle.com)
 (PORT=32044))
 (ADDRESS=
 (PROTOCOL=TCP)
 (HOST=host2-vip.us.oracle.com)
 (PORT=32044)))
 (CONNECT_DATA=
 (SERVER=DEDICATED)
 (SERVICE_NAME=beehive.us.oracle.com)))
```

---

## Installing Oracle Beehive in High Availability Environment

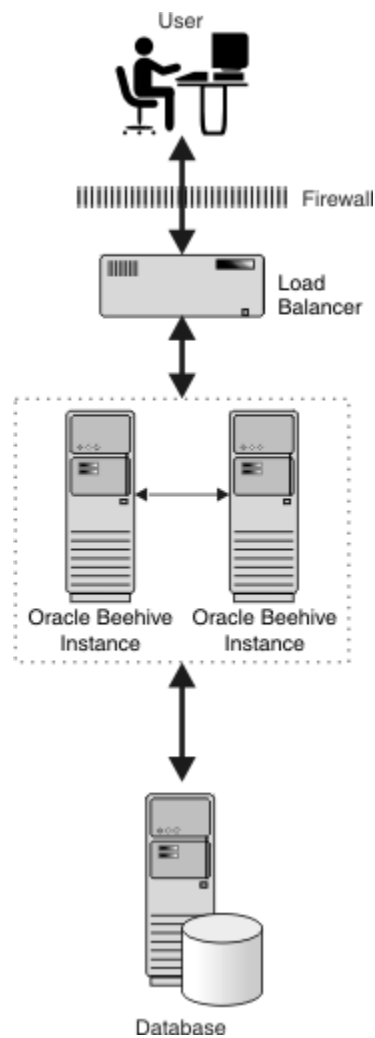
Installing Oracle Beehive in a high availability environment involves installing a third-party load balancer or an Oracle Beehive DMZ instance (or both) and configuring the virtual server of each Oracle Beehive (non-DMZ) instance. Configuring the virtual server ensures that there is a single point of access to your Oracle Beehive instances.

This chapter 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 SSL Termination at 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 33–1 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	
<b>HttpPort</b>	<b>7777</b>
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:** 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

---

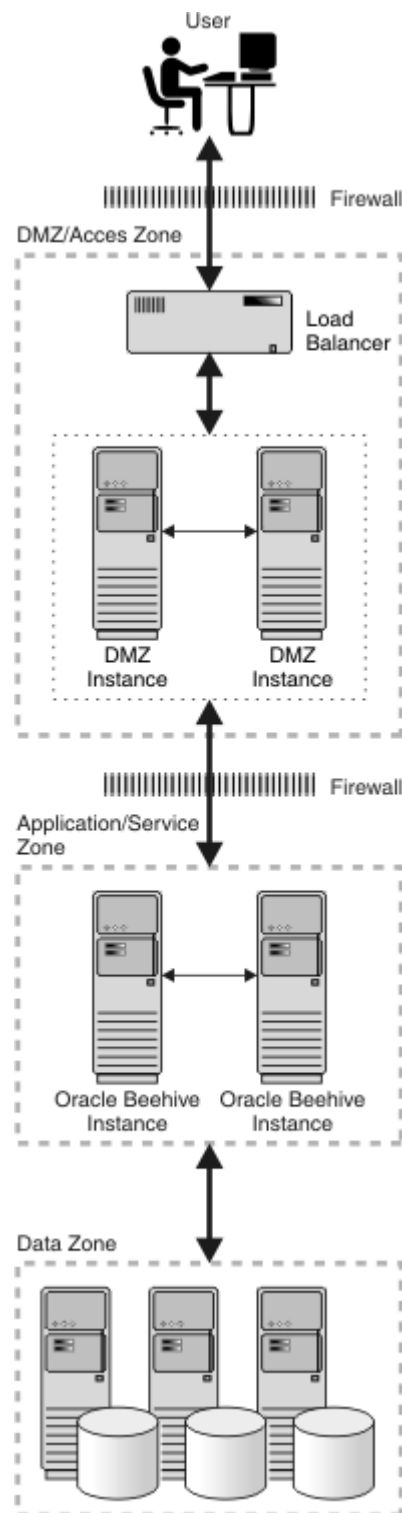
**Note:** If you are planning to install Oracle Beekeeper in a DMZ-based environment, then install Oracle Beekeeper in an Oracle Beehive application tier in a new Oracle home.

Oracle Beekeeper cannot be accessed from a DMZ instance. It should only be accessed from the computer in which you installed Oracle Beekeeper.

---

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 33-2 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:** 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 HTTP 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 2-3 for each of your Oracle Beehive DMZ instances.

## Configuring SSL Termination at Load Balancer

If your load balancer supports SSL termination or offloading, you may offload SSL processing to your load balancer so that your Oracle Beehive instances do not have to decrypt SSL-encrypted data, thereby reducing the load of your Oracle Beehive instances.

You may offload SSL processing to your load balancer for any protocol supported by your load balancer, except if you wish to use the ability of BTP clients, such as Oracle Beehive Extensions for Outlook, to tunnel its connections through HTTPS. In that case, you cannot offload SSL for HTTPS to your load balancer, but you could offload any other protocol.



---

**Note:** Tunneling is the encapsulation of one protocol with another different protocol. This is useful if you are using a BTP client behind a firewall that does not allow BTP(s) connections. In particular, tunneling allows Oracle Beehive Extensions for Outlook to connect through this firewall; Oracle Beehive Extensions for Outlook can encapsulate its BTP(s) session so that it appears to the firewall to be a regular HTTPS session.

---

After configuring your load balancer with your Oracle Beehive deployment, follow these steps to configure Oracle Beehive for SSL termination:

1. Set the `SslTerminatedByLoadBalancer` property of the `HttpServerCluster` component to true:

```
beectl modify_property
--component _current_site:HttpServerCluster
--name SslTerminatedByLoadBalancer
--value true
--activate_configuration
```

2. Commit changes made to the configuration:

```
beectl modify_local_configuration_files
```

If your Oracle Beehive deployment is configured with SSL Termination at the load balancer, follow the steps below to enable HTTPS tunneling:

1. Register a new hostname (for example, `beehive-mx.example.com`)
2. Generate a Certificate Request to include the new hostname in the Beehive certificates. Refer ["Configuring SSL"](#) for more information on certificates.

---

**Note:** It is recommended that the primary common name on the certificate must match the primary client-facing server name. The certificate must match both the primary and MX tunneling hostnames. For example `beehive.example.com` and `beehive-mx.example.com`. The certificates must use the **Subject Alternative Name** field and specify both hostnames.

---

3. Configure the load balancer to not SSL Terminate the traffic to port 443 of `beehive-mx.example.com`, instead redirect the traffic to the https port of the Beehive DMZ instance(s).
4. Adjust properties `MxTunnelPort` and `MxTunnelHostName` to point to the new hostname:

```
beectl modify_property
--component _VIRTUAL_SERVER
--name MxTunnelHostName
--value beehive-mx.example.com
beectl modify_property
--component _VIRTUAL_SERVER
--name MxTunnelPort
--value 443
beectl activate_configuration
```



## Numerics

---

- 64-bit, 1-6
  - 32-bit shell emulation, 1-8
  - required packages, 1-6

## A

---

- actionable notifications
  - post-installation, 15-7
- affinity service, 32-4
- AJPS
  - See* Apache JServ Protocol
- AJSP
  - post-installation, 15-6
- Apache JServ Protocol, 31-1
- ASK user
  - post-installation, 15-7

## B

---

- beeadmin, 4-13
- beectl, 15-2

## C

---

- certificate authority, 21-1
- changing ports, 15-3
- checklist, 3-1
- cloning, 1-15, 15-9, 29-1
  - application tier, 29-7
  - customizing files or directories, 29-19
  - customizing ports, 29-20
  - introduction, 29-1
  - LDAP synchronization, 29-21
  - multiple instances, 29-25
  - OC4J, 29-18
  - Oracle Beehive Integration for Zimbra, 29-19
  - other documentation, 29-25
  - ports, 29-20
  - preparing source application tier, 29-4
  - replicating LDAP server, 29-22
  - site, 29-12
  - SSL, 29-21
  - testing replicated LDAP, 29-24
- code schema, 11-15
- configuring SSL termination, 33-6

## D

---

- data schema, 11-15
- database, 2-1, 11-2
  - admin user name, 4-9
  - admin user password, 4-9
  - creating, 2-5
  - customizing tablespaces, 2-5
    - deleting, 9-5
  - deleting tablespaces and datafiles, 9-3
  - deployment procedure, 6-5
  - general requirements, 2-1
  - host and port, 4-8, 5-4
  - initialization parameters, 2-3
  - install wizard, 4-8
  - logging settings, 2-5
  - minimum tablespace sizes, 2-6
  - Oracle Real Application Clusters, 2-5, 32-1
    - installation, 32-2
    - post-install steps, 32-2
    - pre-install steps, 32-2
  - patches, 2-2
  - post-installation, 15-2
  - raw storage, 2-6
  - repository database, 4-16
  - requirements, 1-9
  - schema user password, 5-4
  - service name, 4-9, 5-4
  - upgrading, 2-6
- deployment procedure, 6-1
  - database information, 6-5
  - deployment targets, 6-3
  - enterprise information, 6-6
  - introduction, 6-1
  - Oracle Beekeeper, 6-8
  - running, 6-2
  - upload, 6-4
  - uploading software image archive, 6-11
- deployment targets, 6-3
- deployment templates, 11-14
- Device Management Service, 25-1, 26-1
- disk space, 1-9
  - Oracle Beekeeper, 1-13
- DMZ, 18-1
  - configuring, 18-2
  - configuring SSL, 19-5

- deployment procedure, 6-1
- high availability, 33-3
- high availability with load balancer, 33-4
- install wizard, 4-3
- installing, 1-11, 4-18
- installing in silent mode, 7-5
- manually deleting, 18-5
- memory, 1-9
- Oracle Wallet, 18-1
- post-installation, 15-5
- troubleshooting, 18-5

## E

---

- e-mail
  - configuring e-mail client, 22-6
  - configuring with SSL, 22-1
  - post-installation, 15-6
- end users
  - Oracle Beehive Extensions for Outlook
    - verifying availability, 25-24
- enterprise, 4-10
- exporting configuration data, 10-1, 11-4

## F

---

- FTP, 17-1
  - enabling active mode FTP, 17-1
  - opening ports, 17-2
  - passive mode FTP, 17-4
  - setting parameters, 17-6

## G

---

- Grid Control
  - installing provisioning application, 1-11
  - requirements, 1-9

## H

---

- hasbind, 11-18
- high availability, 33-1
  - configuring with load balancer, 33-1
  - DMZ, 33-3
  - DMZ and load balancer, 33-4
- host name, 1-10
- HTTP
  - configure Apache HTTP server for WebDAV folders, 21-3

## I

---

- install wizard, 4-1
  - configuration assistants, 4-15, 4-20, 5-5
  - database information, 4-8, 5-4
  - enterprise definition information, 4-10
  - existing enterprise information, 4-12
  - Oracle Beekeeper, 5-1
  - prerequisite checks, 4-5, 4-19, 5-3
  - select installation type, 4-6
  - software image archive, 4-17

- software library configuration, 4-17
- starting, 1-14
- template information, 4-9
- upgrading, 13-1
- installation
  - scenarios, 1-10
  - verifying, 7-5
- installing
  - 64-bit, 1-6
  - non-SSL site, 19-8
  - Oracle Beehive Conferencing client, 27-1
  - Oracle Beehive Extensions for Explorer, 26-1
  - Oracle Beehive Extensions for Outlook, 25-1
  - Oracle Beekeeper, 5-1
  - scenarios
    - against existing database, 1-10
    - DMZ, 1-11, 4-18
    - high availability environment, 1-13, 33-1
    - more than one instance on the same machine, 1-13
    - Oracle Beehive Integration for Zimbra, 1-11
    - Oracle Beekeeper, 1-13
    - provisioning application, 1-11, 4-16
    - silent mode, 1-13
    - standard, 1-10, 4-4
    - verifying, 7-3, 7-5
- instance name, 4-11
- Internet Protocol version 6, 1-10
- inventory location, 4-2, 8-2
  - clones, 29-20
- IPv6, 1-10

## J

---

- job system, 6-2

## K

---

- kernel parameters, 1-3

## L

---

- LDAP
  - post-installation, 15-5
- LDAP synchronization
  - cloning, 29-21
  - configuring SSL, 21-1
  - testing replicated LDAP in cloned instance, 29-24
  - troubleshooting replicated LDAP, 29-25
- load balancer, 33-1, 33-6

## M

---

- memory, 1-9
  - Oracle Beekeeper, 1-13
- Microsoft Active Directory
  - installing Oracle Beehive Extensions for Outlook
    - verifying availability for end users, 25-24
- mobile device management service
  - upgrading, 11-18
- multiple instances, 1-13, 20-3

Oracle Real Application Clusters, 32-4

## O

---

oc4jadmin, 4-13

operating system group, 4-3

Oracle Application Server Certificate  
Authority, 21-1

Oracle Beehive

deployment procedure, 6-1

restarting OC4J, 30-8

virtual host, 30-7

Oracle Beehive Coexistence Connector  
provisioning application, 6-9

Oracle Beehive Conferencing client, 27-1

installing

Device Management Service based, 27-3

remote downloader, 27-3

system requirements, 27-1

uninstalling, 27-5

updating, 27-4

Oracle Beehive Extensions for Explorer, 26-1, 26-2

installing

desktop based, 26-8

Device Management Service based, 26-2

language pack, 26-7

system requirements, 26-1

uninstalling, 26-8, 26-12

updating, 26-6

upgrading, 26-10

Oracle Beehive Extensions for Outlook, 25-1, 25-2

configuration wizard, 25-17

configuring, 25-6

customizing

installation settings, 25-10

installing

Active Directory, 25-22

desktop based, 25-14

Device Management Service based, 25-2

language pack, 25-7

lockdown systems, 25-15

migration tool, 25-24

system requirements, 25-1

uninstalling, 25-14, 25-22

updating, 25-6

upgrading, 25-21

Oracle Beehive Integration for Zimbra, 1-11, 11-4,  
28-1

cloning, 29-19

HTTP port, 28-2

performance, improving, 28-3

post-installation, 15-8, 28-1

screen resolution, 28-1

SSL, 28-1

Oracle Beekeeper, 1-13

changing port, 30-3

changing RMI port, 30-3

deployment procedure, 6-8

extending Java Single Sign-On session  
timeout, 30-3

installing, 5-1

LDAP-based authentication, 30-4

Oracle Wallet, 30-2

post-installation, 15-10, 30-1

software image archive, 4-17

SSL, 30-1

upgrading, 11-10, 13-1

Oracle home, 4-4

Oracle Management Service, 4-16

Oracle RAC

enabling affinity service, 32-4

Oracle Real Application Clusters, 2-5, 32-1

installation, 32-2

multiple instances, 32-4

post-install steps, 32-2

pre-install steps, 32-2

Oracle uninstalling

Beehive Extensions for Outlook, 25-22

Oracle Wallet, 20-3

auto-login, 20-1

changing password, 20-2

configuring, 20-2

configuring TLS, 20-1

configuring with Oracle Beekeeper, 30-2

creating CA-signed certificate and importing  
it, 19-7

creating self-signed certificate, 19-7

creating self-signed certificate and importing  
it, 19-6

DMZ, 18-1

multiple instances, 20-3

ORMIS (ORMI over SSL), 20-3

disable, 20-4

enable with default SSL, 20-4

enable with password protection, 20-4

oraInventory, 4-2, 8-2

organization, 4-10

Outlook-post-installation, 15-8

## P

---

port numbers

changing HTTP port, 15-2

listen on ports less than 1024, 15-2

ports

adding endpoint, 22-3, 22-5

changing, 15-3

changing HTTP port, 15-2

cloning, 29-20

creating and configuring endpoint, 22-5

customizing ports in cloned instance, 29-20

enabling non-SSL and SSL ports, 22-2

FTP, 17-2

HTTP port, Oracle Beehive Integration for  
Zimbra, 28-2

less than 1024, 15-2

modifying endpoint, 22-2, 22-5

post\_upgrade\_db\_actions.pl, 11-12

post-installation, 1-14, 15-1

actionable notifications, 15-7

- AJPS, 15-6
- ASK user, 15-7
- certificate authority, 21-1
- cloning, 29-1
- configuring e-mail with SSL, 22-1
- configuring Oracle Beehive Integration for Zimbra, 28-1
- database, 15-2
- DMZ, 15-5, 18-1
- e-mail, 15-6
- enabling AJPS, 31-1
- LDAP, 15-5
- listen on ports less than 1024, 15-2
- Oracle Beehive Integration for Zimbra, 15-8
- Oracle Beekeeper, 15-10, 30-1
- Oracle Wallet, 20-1
- Outlook, 15-8
- remote media server, 15-9
- Secure Enterprise Search
  - Secure Enterprise Search
    - post-installation, 15-7
- Single Sign-On, 15-7
- SMS notifications, 15-7
- SSL, 15-6, 19-1
- TLS, 15-6, 20-1
- XMPP, 15-7
- preparing for upgrade, 11-4
- prerequisites, 1-1
- preview mode, 4-1
- privileged port numbers, 15-2
- provisioning advisor framework, 6-1
- provisioning application, 1-11
  - install wizard, 4-3
  - installing in silent mode, 7-4
- Oracle Beehive Coexistence Connector, 6-9

## R

---

### RAC

- See* Oracle Real Application Clusters
- remote downloader, 25-2, 26-2
- remote media server
  - post-installation, 15-9
- requirements, 1-1
  - database, 1-9, 2-1
    - general requirements, 2-1
    - initialization parameters, 2-3
    - logging settings, 2-5
    - patches, 2-2
  - hardware, 1-9
  - host name resolution, 1-10
  - kernel parameters, 1-3
  - kernel version, 1-2
  - operating system, 1-2
  - Oracle Beekeeper, 1-13
  - Oracle Enterprise Manager Grid Control, 1-9
  - packages
    - Linux, 1-4
  - shell limits, 1-8
- response file, 7-1, 7-2, 7-3, 8-1, 8-2

- creating, 7-2, 7-3

## S

---

- schema password, 4-12
- setting shell limits, 1-8
- shell limits, 1-8
- silent mode, 1-13, 7-1
  - DMZ, 7-5
    - provisioning application, 7-4
    - uninstalling, 7-6
    - upgrading, 7-5
- Single Sign-On, 24-1
  - post-installation, 15-7
- site, 4-11
- site key, 4-11, 5-4
- SMS notifications
  - post-installation, 15-7
- software image archive, 4-17
- software library, 6-2
- software library location, 4-17
- SSL, 20-1
  - checklist, 19-1
  - cloning, 29-21
  - configuring, 19-1
    - Apache HTTP server for WebDAV, 21-3
    - DMZ, 19-5
    - LDAP integration, 21-1
    - self-signed certificates, 19-3, 19-4
    - test certificates, 19-2
  - configuring termination at load balancer, 33-6
  - configuring with Oracle Beekeeper, 30-1
  - installing non-SSL Oracle Beehive site, 19-8
  - LDAP, configure for, 21-1
  - post-installation, 15-6
- swap space, 1-9

## T

---

- templates, 4-9, 6-7
- TLS, 20-1
  - post-installation, 15-6
- tnsnames.ora file, 11-3
- troubleshooting, 10-1
  - automated failover, 32-6
  - DMZ configuration, 18-5
  - exporting configuration data, 10-1
  - recovering from failed configuration, 7-5
  - replicated LDAP, 29-25

## U

---

- uninstall, 9-1
  - command-line option, 8-3
  - deleting customized tablespaces and datafiles, 9-5
  - deleting tablespaces and datafiles, 9-3
  - multiple instances, 9-3
  - Oracle Beekeeper, 5-6
  - starting Oracle Beekeeper uninstall wizard, 5-6
- uninstall wizard, 5-6

- uninstalling, 1-15
  - corrupted installation, 9-5
  - Oracle Beehive Conferencing client, 27-5
  - Oracle Beehive Extensions for Explorer, 26-8, 26-12
  - Oracle Beehive Extensions for Outlook, 25-14
  - provisioning application, 9-2
  - silent mode, 7-6
- upgrading, 1-14, 11-1, 11-2, 11-3, 12-1
  - analyzing application tiers, 11-3
  - applying deployment template, 11-14
  - before you start, 11-2
  - clone patching, 14-1
    - types, 14-2
  - database, 2-6, 11-2
  - gathering statistics, 11-15
  - hasbind, 11-18
  - mobile device management service, 11-18
  - multiple application tiers, 11-10
  - Oracle Beehive Integration for Zimbra, 11-4
  - Oracle Beekeeper, 11-10, 13-1
  - post\_upgrade\_db\_actions.pl, 11-12
  - rolling back Application Server Critical Patch Update, 11-3
  - sequence, 11-9
  - shutting down instances, 11-9
  - silent mode, 7-5
  - user directory services, 11-17
  - User Directory Services synchronization
    - enabling, 11-16
  - voicemail, 11-15
  - XmppTimerKeepAliveTime, 11-2
  - Zero Downtime Upgrade, 11-6
- user directory services
  - upgrading, 11-17

## **X**

---

### **XMPP**

- post-installation, 15-7

## **Z**

---

- Zero Downtime Upgrade, 11-6

