

Oracle® Beehive
Deployment Guide
Release 1 (1.2 and 1.3)
E10476-02

August 2008

Oracle Beehive Deployment Guide, Release 1 (1.2 and 1.3)

E10476-02

Copyright © 2008, Oracle. All rights reserved.

Primary Author: Joe Paradise

Contributing Authors: Raymond Gallardo, Valarie Moore

Contributors: Henrik Blixt, Vimal Chopra, Ramesh Dommeti, Brennan Gaunce, Richard Hall, Duane Jensen, Tait McCarthy, Andrew Mitchell, Paul Nock, Mark Paterson, Francois Perrault, Jay Rajiva, Jamie Rancourt, Reza Rokni, Sudip Roy, Josh Stanley

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

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

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

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

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

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

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

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	viii
Conventions	viii
1 Overview of Deploying Oracle Beehive	
Overview of Oracle Beehive	1-1
How to Use This Guide to Deploy Oracle Beehive	1-1
Prerequisites for Deploying Oracle Beehive	1-2
Deploying Oracle Beehive in Stages	1-2
Oracle Beehive Deployment Stage #1: Create a Deployment Plan	1-3
Oracle Beehive Deployment Stage #2: Research and Knowledge Gathering.....	1-3
Oracle Beehive Deployment Stage #3: Requirements Gathering	1-4
Oracle Beehive Deployment Stage #4: Install and Configure the System	1-4
Oracle Beehive Deployment Stage #5: Customize the System	1-4
Oracle Beehive Deployment Stage #6: Evaluate and Test the System.....	1-5
Oracle Beehive Deployment Stage #7: Training for Developers, Administrators, and End Users 1-5	
Oracle Beehive Deployment Stage #8: Launch the System into Production	1-5
Oracle Beehive Deployment Stage #9: On-going Maintenance and Administration.....	1-5
Key Terms and Concepts for Deploying Oracle Beehive	1-6
Servers and Nodes	1-6
Network Zones	1-6
Demilitarized Zones (DMZs).....	1-6
Installations	1-6
Services	1-6
Instances	1-7
Sites	1-7
Enterprises and Organizations.....	1-7
2 Oracle Beehive Architecture	
Overview of the Oracle Beehive Tiers	2-1
Overview of the Client Tier	2-2
Overview of the Application Tier	2-2

Overview of the Data Tier.....	2-3
Overview of the Ancillary Tier	2-3
Connections Between Oracle Beehive Tiers	2-4
Overview of the Database Access Framework	2-4
Overview of the Beehive Transport Infrastructure	2-4
Overview of the Event Framework	2-5
Overview of Oracle Beehive Schemas.....	2-5
Code Schema	2-6
Data Schema	2-6
3 Tools for Deploying Oracle Beehive	
The Oracle Beehive Install Wizard	3-1
The Oracle Beehive Config Wizard	3-1
Oracle Beehive Administration Console	3-2
The Oracle Beehive <code>beectl</code> Utility.....	3-2
Oracle Enterprise Manager Grid Control	3-2
4 Deploying Oracle Beehive with Other Software Components	
Deploying Oracle Beehive with Oracle Database	4-1
Deploying Oracle Beehive with Oracle Real Application Clusters (Oracle RAC).....	4-2
Preparing to Deploy Oracle Beehive with Oracle RAC.....	4-2
Post-installation Steps for Deploying Oracle Beehive with Oracle RAC.....	4-3
Deploying Oracle Beehive with an External User Directory.....	4-3
Preparing to Deploy Oracle Beehive with an External User Directory.....	4-4
Limitations of Deploying Oracle Beehive with an External User Directory	4-4
Deploying Oracle Beehive with Oracle Universal Records Management (URM)	4-5
Oracle Beehive-based Tasks for Deploying Oracle Beehive with Oracle URM.....	4-5
Oracle URM-based Tasks for Deploying Oracle Beehive with Oracle URM	4-5
Deploying Oracle Beehive with Oracle Wallet.....	4-6
Deploying Oracle Beehive with an External Oracle BPEL Process Manager Instance.....	4-6
Deploying Oracle Beehive with Oracle Secure Enterprise Search 10g.....	4-8
Deploying Oracle Beehive with Microsoft Exchange Server 2003	4-8
Installing and Configuring the Oracle Collaboration Coexistence Gateway	4-9
Load Balancer Requirements for Deploying Oracle Beehive with Microsoft Exchange Server 2003	4-9
Deploying Oracle Beehive with Symantec Scan Engine.....	4-10
5 Network Considerations for Oracle Beehive	
Basic Network Considerations	5-1
Network Considerations for Small Deployments	5-1
Access and Security Considerations	5-1
Traffic Considerations	5-2
Internal Traffic	5-2
External Traffic	5-2
Other Network Considerations for Deploying Oracle Beehive.....	5-2

6	Architectural Considerations for Oracle Beehive	
	Basic Architectural Considerations.....	6-1
	Access Considerations	6-1
	Capacity Considerations	6-2
	Recovery Considerations	6-2
	Availability Considerations	6-2
	Scalability Considerations.....	6-2
	Workload Considerations	6-2
	Security Considerations	6-3
	Organization-specific Architecture Considerations	6-3
7	Oracle Beehive Deployment Configurations	
	Deploying Oracle Beehive on a Single Computer.....	7-1
	Deploying Oracle Beehive on Multiple Computers.....	7-3
	Deploying Oracle Beehive Across Network Zones	7-4
	How the Beehive Transport Infrastructure (BTI) Supports Multiple Network Zones	7-6
8	Deploying Oracle Beehive with Supported Clients and Devices	
	Oracle Beehive Deployments with Oracle Beehive Integration for Outlook (OBIO).....	8-1
	Oracle Beehive Deployments with Oracle Beehive Integration for Zimbra	8-2
	Oracle Beehive Deployments with Standards-based Clients.....	8-2
	Oracle Beehive Deployments with Mobile Devices	8-3
	Deploying Microsoft Windows Mobile-based Clients Supported by Oracle Beehive.....	8-4
	Enabling the Oracle Beehive Short Message Service (SMS) Delivery Channel	8-4

Index

List of Figures

2-1	Oracle Beehive Logical Architecture.....	2-2
7-1	Oracle Beehive and Oracle Database Deployed on a Single Computer	7-2
7-2	Oracle Beehive and Oracle Database Deployed on Different Computers.....	7-3
7-3	Multiple Oracle Beehive Instances on Multiple Computers.....	7-4
7-4	Oracle Beehive Deployed Across Multiple Network Zones.....	7-5

Preface

This guide describes the concepts, considerations, and options associated with deploying Oracle Beehive.

This preface contains the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This guide is intended for anyone who is involved with deploying Oracle Beehive. This guide is particularly useful to administrators and other resources who plan to deploy and manage Oracle Beehive.

Documentation Accessibility

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

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

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

Related Documents

For more information, refer to the following documents in the Oracle Beehive documentation library:

- *Oracle Beehive Administrator's Guide*
- *Oracle Beehive Administrator's Reference Guide*
- *Oracle Beehive Application Developer's Guide*
- *Oracle Beehive Concepts*
- *Oracle Beehive End-User Information*
- *Oracle Beehive Installation Guide for Linux*
- *Oracle Beehive Installation Guide for Microsoft Windows*
- *Oracle Beehive Installation Guide for Solaris Operating System (SPARC 64-Bit)*
- *Oracle Beehive Licensing Information*
- *Oracle Beehive Release Notes*
- *Oracle Beehive Administration Console Online Help*

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.

Overview of Deploying Oracle Beehive

This module provides an introduction to Oracle Beehive and an overview of the deployment process, and includes the following topics:

- [Overview of Oracle Beehive](#)
- [How to Use This Guide to Deploy Oracle Beehive](#)
- [Prerequisites for Deploying Oracle Beehive](#)
- [Deploying Oracle Beehive in Stages](#)
- [Key Terms and Concepts for Deploying Oracle Beehive](#)

Overview of Oracle Beehive

Oracle Beehive is the next generation of Oracle's enterprise collaboration technologies. Oracle Beehive provides a unified collaboration application and platform comprised of a comprehensive set of integrated, Java-based services, offering a new paradigm for enterprise collaboration solutions.

Oracle Beehive unifies common yet typically disjointed collaborative services, such as e-mail, time management, instant messaging, and content management, among others, and delivers them through standard protocol clients, integrated Web services, and familiar desktop tools such as Microsoft Outlook. This unified offering enables teams, individuals, and organizations to more easily collaborate, generate new information, make timely decisions, and, ultimately, take action.

Oracle Beehive allows IT departments to consolidate their collaboration-enabling infrastructures and implement people-centric, rather than tool-centric, applications in a centrally managed, secure, and compliant environment. The Oracle Beehive platform is built on proven and cost-effective Oracle technologies, such as Oracle Database and Oracle Application Server, which provide reliability, manageability, scalability, and performance.

How to Use This Guide to Deploy Oracle Beehive

Oracle Beehive is a comprehensive enterprise collaboration platform that provides a wide range of functionality for deployments both small and large. As with any complex software technology that is designed to support the needs of an entire enterprise, there are many options to consider related to its deployment. Deploying Oracle Beehive can be complex from the standpoint that the system can leverage and interoperate with many hardware and software components. However, with proper planning as well as the right tools and resources in place, deploying Oracle Beehive can be straightforward and easily achieved.

This guide provides details on the things you'll need to consider when planning to deploy Oracle Beehive. This guide will also help you evaluate many aspects of your organization's deployment needs, and will enable you to choose and plan the most appropriate path for successfully and optimally deploying Oracle Beehive. For reference purposes, examples of common deployment scenarios are provided. References and links to more in-depth details on certain options and components, especially those provided by other Oracle products, are also provided where appropriate.

Note: To gain a comprehensive, high-level understanding of the features, options, and benefits that Oracle Beehive provides, you should first read *Oracle Beehive Concepts*, if you have not already done so.

This guide provides details on the various aspects for consideration when deploying Oracle Beehive, including:

- Network considerations
- Architectural considerations
- Integration with other supported Oracle and third-party products
- High availability
- Coexistence with third-party components
- Installation and upgrades

Note: No install- or upgrade-related procedural information is provided in this guide. For that type of information, please refer to the *Oracle Beehive Installation Guide* for your operating system.

- Deploying supported clients
- Supported deployment scenarios

Prerequisites for Deploying Oracle Beehive

To successfully deploy Oracle Beehive you should know or understand the following:

- Oracle Beehive architecture
- Available Oracle Beehive features and options
- Your organization's deployment needs
- The resources that are available for deployment, including software and hardware components as well as technical personnel

You should read this and other related chapters prior to installing Oracle Beehive and allowing users to access the system.

Deploying Oracle Beehive in Stages

Prior to deploying Oracle Beehive, it is strongly recommended that you consider a framework and methodology to follow through the various deployment stages. Doing

so will help you and your organization prepare for, plan, and execute a successful deployment of Oracle Beehive.

For your convenience, a sample framework to follow is provided in this section. However, following this framework is not mandatory for a successful deployment of Oracle Beehive. You should use whatever approach or methodology that best meets the needs, standards, and obligations of your organization based on the available resources and other factors.

The framework provided here is divided into several stages that follow each other in order. This framework also integrates the use of three successive deployment environments: the first for development or customization, the second for testing, and the third for production. Following this or a similar framework will help ensure a successful initial deployment of Oracle Beehive. It will also enable your organization to successfully maintain, administer, and upgrade the system, as needed, going forward.

The proposed framework for deploying Oracle Beehive includes the following stages, in order:

- [Oracle Beehive Deployment Stage #1: Create a Deployment Plan](#)
- [Oracle Beehive Deployment Stage #2: Research and Knowledge Gathering](#)
- [Oracle Beehive Deployment Stage #3: Requirements Gathering](#)
- [Oracle Beehive Deployment Stage #4: Install and Configure the System](#)
- [Oracle Beehive Deployment Stage #6: Evaluate and Test the System](#)
- [Oracle Beehive Deployment Stage #7: Training for Developers, Administrators, and End Users](#)
- [Oracle Beehive Deployment Stage #8: Launch the System into Production](#)
- [Oracle Beehive Deployment Stage #9: On-going Maintenance and Administration](#)

Oracle Beehive Deployment Stage #1: Create a Deployment Plan

The first thing that you and others in your organization should do is choose an approach or methodology for deploying Oracle Beehive. The second thing that you should do is put that approach in writing and begin to create a plan based on it. The initial plan that you create should contain the general or high-level tasks that you expect will be required to deploy your Oracle Beehive instance. This can include major stages such as those listed here. The addition of specific, low-level tasks and their details should be added to the plan only as they become known, which often occurs in the following stages of the deployment process.

You should also begin to identify key resources that will be involved in the deployment process. These resources may include system administrators and technical staff members as well as senior-level members in your organization who possess decision-making and signoff authority.

Oracle Beehive Deployment Stage #2: Research and Knowledge Gathering

The research and knowledge gathering stage includes gaining a solid understanding of what Oracle Beehive is as well as what related options are available to you and your organization. To gain a comprehensive, high-level understanding of the features, options, and benefits that Oracle Beehive provides, you should first read *Oracle Beehive Concepts* if you have not already done so. Then, you should read this guide.

During this stage, you should gain an understanding of how your organization plans to deploy and use Oracle Beehive. You should understand your organization's goals for deploying Oracle Beehive, and what the motivators and constraints are for achieving those goals, including any time frames or deadlines. This should also include confirming whether or not your organization plans to migrate users and data from existing systems. Remember to consider your organization's current and future needs, whether actual or potential, as you gather this information.

Oracle Beehive Deployment Stage #3: Requirements Gathering

Once you have a better understanding of the various available Oracle Beehive options and components, you should determine which ones will apply to your deployment and then gather the information related to their requirements. This includes obtaining hardware and software requirements, server and host names, and port numbers, among other important information. This may also include identifying users and data sets that will be migrated from existing systems to Oracle Beehive, and how your organization plans to migrate them.

Tip: To ease the transition from a development environment to a production environment and beyond, it is best to use virtual host names from the beginning. With this approach, you can transition and grow the system without having to significantly modify existing integrations of servers, services, and other components.

This guide provides general deployment considerations and recommendations as they pertain to system requirements. For specific hardware and software requirements as well as other installation-specific information, refer to *Oracle Beehive Installation Guide* for your operating system.

Oracle Beehive Deployment Stage #4: Install and Configure the System

The installation stage includes all installation-related steps, including pre- and post-installation steps. Pre-installation steps include, among others, mapping out and planning for the installation process, preparing your operating system, and verifying that Oracle Database is already installed. Another pre-installation step is obtaining the Oracle Beehive installation package.

Oracle provides the Oracle Beehive installation package through an installation medium (CD, DVD, and so on) or through a download process from a supported Oracle website. For more information, refer to the *Oracle Beehive Installation Guide* for your operating system.

Post-installation steps include verifying that the installation was successful, configuring fundamental aspects of the system, and, optionally, beginning to migrate data from existing systems.

Oracle Beehive Deployment Stage #5: Customize the System

This stage includes customizing Oracle Beehive in one or more ways such as by integrating the system with other supported Oracle and third-party applications. This stage may also include deploying Oracle Beehive into a development environment to customize the system programmatically. This can include extending Oracle Beehive services or leveraging the platform's application programming interface (API). The Oracle Beehive API enables organizations to retrieve and manipulate Oracle Beehive data as well as integrate the system with custom and third-party applications. For more information, refer to the *Oracle Beehive Application Developer's Guide*.

Oracle Beehive Deployment Stage #6: Evaluate and Test the System

After installing and customizing Oracle Beehive, but prior to launching the system, it is critical that you or your organization rigorously test the system to ensure that it functions as expected. This includes testing various server configurations and custom development, tuning the system, verifying that data from existing systems has migrated successfully, and conducting functional tests on all end-user clients that will be deployed into production. This stage also includes making any necessary changes if issues are discovered.

Note: For more information on configuring and tuning the system, refer to the *Oracle Beehive Administrator's Guide*.

Test environments can either be replicas of their associated production environments (recommended) or they may be scaled-down versions that mimic production environment topologies but with less hardware.

Oracle Beehive Deployment Stage #7: Training for Developers, Administrators, and End Users

An often-overlooked, but fundamentally important aspect of deploying a new system or application is training. In this stage, developers, administrators, and end users receive the information and instructions that they will need to begin customizing, managing, and collaborating with Oracle Beehive. This can include in-class training, online tutorials and demos, or other formats that are appropriate for your organization and the resources that are available. Oracle Beehive supports a variety of clients and users of all levels, so it might make sense to consider a different training strategy and delivery medium for each client or user type.

Including this stage in your overall deployment strategy will benefit your organization on many levels. For example, providing adequate training will likely reduce end-user issues and support costs. Conversely, it will also likely boost end-user adoption and productivity, ensuring that your organization receives the maximum return on the benefits provided by Oracle Beehive.

Oracle Beehive Deployment Stage #8: Launch the System into Production

After testing Oracle Beehive and training end users, the next stage is to launch system into a production environment, or "go live". This is the stage where all of the work and preparations from the previous stages come to fruition. Steps such as using virtual host names are of particular importance and can greatly simplify the process of launching the system. Depending on the protocols and needs of your organization, launching the system into production might include rolling out to one or more selected user groups at first and then, after confirming system stability, to the remainder of the enterprise. This stage may also include migrating users and data from existing systems.

Oracle Beehive Deployment Stage #9: On-going Maintenance and Administration

After launching the system, you will enter the on-going maintenance and administration stage. This stage includes monitoring Oracle Beehive, changing configuration settings as needed, and managing users and groups, among other administrative tasks. This may also include coordinating regular updates and upgrades for the deployment of additional features, services, or hardware. For more information, refer to the *Oracle Beehive Administrator's Guide*.

Key Terms and Concepts for Deploying Oracle Beehive

Before beginning the deployment process for Oracle Beehive, it is important to understand several fundamental deployment terms and concepts, including:

- [Servers and Nodes](#)
- [Network Zones](#)
- [Demilitarized Zones \(DMZs\)](#)
- [Installations](#)
- [Services](#)
- [Instances](#)
- [Sites](#)
- [Enterprises and Organizations](#)

Servers and Nodes

A server is a computer on which an Oracle Beehive instance runs. A node is a server in a network deployment of Oracle Beehive. Oracle Beehive supports deployments with multiple nodes.

Network Zones

A network zone is a defined and segregated area of a network that hosts one or more Oracle Beehive components.

Demilitarized Zones (DMZs)

A demilitarized zone (DMZ) is a computer host or small network that an enterprise inserts as a neutral zone between its internal, private network and the Internet. A DMZ prevents unauthorized users from accessing systems that contain confidential or proprietary data. In effect, DMZs act as proxy servers and provide another layer of network security, especially when used with firewalls.

Installations

An installation is the set of bits that comprise Oracle Beehive. Although multiple installations may exist on the same server, typically there is only one installation for each server.

Services

An Oracle Beehive service is a discrete implementation of specific functionality that users and other services can leverage to accomplish a task. The capabilities and interactions of services enable the full scope of functionality that Oracle Beehive provides.

Oracle Beehive services can be installed and deployed on a single computer or among several computers distributed across multiple network zones. In the latter case, when multiple instances of service are deployed, the collection is referred to and represented as a single implementation of that service. By default, all services are deployed during the Oracle Beehive installation process.

Instances

An instance is an Oracle Beehive server that is running on a computer in an Oracle home (ORACLE_HOME). An instance may respond to requests from a specific enterprise. Oracle Beehive supports one server instance on each computer.

Sites

A site is a collection of physical hardware in the same geographic location used to run Oracle Beehive. Each Oracle Beehive site requires a minimum set of components to support a fully-functioning system. However, within each site, multiple instances of a variety of supported components, including services and servers, can be implemented. This can be useful for the purposes of availability and performance. Oracle Beehive currently supports only one site per deployment.

Enterprises and Organizations

An enterprise is a container for all of the users, groups, and resources that leverage or that are a part of a single Oracle Beehive instance. Typically, an enterprise includes all of the users, groups, and resources within a company. Oracle Beehive supports one enterprise for each deployment.

Enterprises are the containers for organizations, which are groups defined by a department, line of business, project, or other criteria.

The hierarchical relationship between enterprises and organizations in Oracle Beehive provides a logical structure for organizing users, groups, and resources. This structure also allows system administrators to more easily manage users, groups, and resources through their organizational associations.

Oracle Beehive Architecture

Built on Java 2 Platform Enterprise Edition (J2EE), Oracle Beehive provides a multitier architecture that leverages proven Oracle technologies, such as Oracle Database and Oracle Application Server, as well as other key Oracle and third-party components.

This module provides a high-level overview of the Oracle Beehive architecture and its supported components, and includes the following topics:

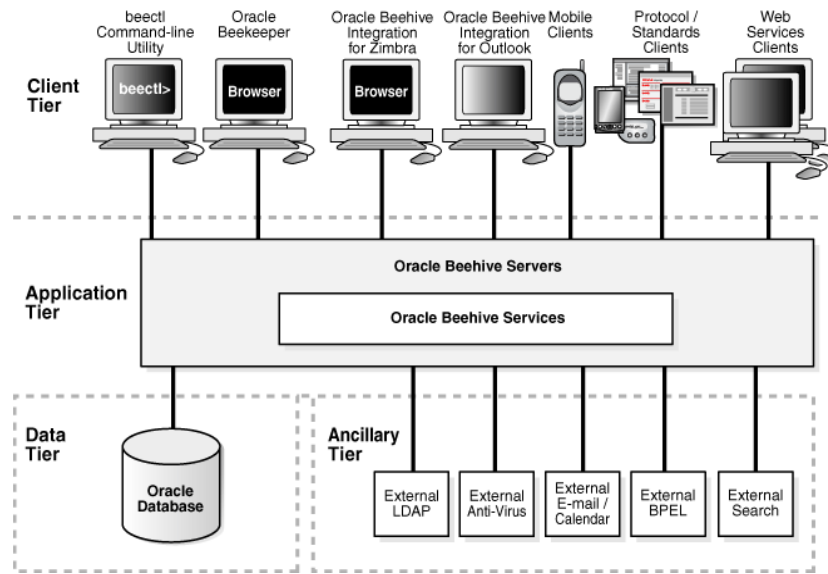
- [Overview of the Oracle Beehive Tiers](#)
- [Connections Between Oracle Beehive Tiers](#)

Overview of the Oracle Beehive Tiers

The Oracle Beehive architecture is divided into the following logical tiers:

- The Client Tier, which contains Oracle Beehive clients, as well as other supported clients and devices. For more information, refer to "[Overview of the Client Tier](#)".
- The Application Tier, which contains Oracle Beehive services and application server components. For more information, refer to "[Overview of the Application Tier](#)".
- The Data Tier, which contains Oracle Database to store system configuration and collaboration data. For more information, refer to "[Overview of the Data Tier](#)".
- The Ancillary Tier, which contains optional Oracle and third-party components that enhance or coexist with certain aspects of the system. For more information, refer to "[Overview of the Ancillary Tier](#)".

Figure 2–1 Oracle Beehive Logical Architecture



Overview of the Client Tier

The Client Tier is the face of the system and includes all supported clients and devices, including integration with Microsoft Outlook and a command-line tool for system administration. Oracle Beehive also supports clients and devices that leverage the following standardized protocols:

- Calendaring Extensions for WebDAV (CalDAV)
- Extensible Messaging and Presence Protocol (XMPP)
- File Transfer Protocol (FTP)
- Internet Message Access Protocol (IMAP)
- Push Internet Message Access Protocol (P-IMAP)
- Open Mobile Alliance Data Synchronization (OMA-DS)
- Simple Mail Transfer Protocol (SMTP)
- Web-based Distributed Authoring and Versioning (WebDAV)

The open nature of the Oracle Beehive platform also enables developers to build and implement custom clients in the Client Tier using Oracle Beehive Web services.

For more information about the deployment options and considerations related to components in the Client Tier, refer to "[Deploying Oracle Beehive with Supported Clients and Devices](#)".

Overview of the Application Tier

The Application Tier is the core of the system and includes all Oracle Beehive server components, including interoperable, function-specific services that provide the system's enterprise collaboration features.

The Application Tier supports multiple Oracle Beehive server instances. Each Oracle Beehive server instance includes required components of Oracle Application Server 10g, which itself hosts the Oracle Beehive services, including:

- **Oracle HTTP Server:** The Web server component of Oracle Application Server 10g. Enables connections between supported clients over HyperText Transport Protocol (HTTP) and Secure HyperText Transport Protocol (HTTPS).
- **Oracle Application Server Containers for J2EE (OC4J):** J2EE v1.4 -compliant containers that provides an infrastructure for deploying, undeploying, and redeploying J2EE-compliant applications and modules. Oracle Beehive services are deployed in OC4J containers.

When you install Oracle Beehive, the required Oracle Application Server 10g components are pre-bundled and installed by default.

Each Oracle Beehive server instance also includes the Beehive Transport Infrastructure (BTI), which enables connectivity between supported clients and Oracle Beehive through its proprietary multiplexor protocol (MX). For more information, please refer to "[Overview of the Beehive Transport Infrastructure](#)".

Overview of the Data Tier

The Data Tier is the information store for Oracle Beehive and contains Oracle Database, either as a single, standalone database instance or an Oracle Real Application Cluster (Oracle RAC). All system configuration and collaborative data for Oracle Beehive is stored in Oracle Database.

The Data Tier provides Oracle Database a layer of separation from the other tiers, ensuring, among other things, optimized security and system performance. Only the Database Access Framework (provided by Oracle Beehive) can access the Data Tier. Oracle Beehive services cannot access the Data Tier directly and must make all connections through the schemas provided by the Database Access Framework. For more information on the Database Access Framework, please refer to "[Overview of the Database Access Framework](#)".

Overview of the Ancillary Tier

The Ancillary Tier contains any optional servers and applications that are external to the Oracle Beehive server. Typically, components in this tier are optional because Oracle Beehive already provides many of these capabilities, such as user directories, e-mail, and time management.

Oracle Beehive supports Ancillary Tier components to provide enterprises flexibility in their deployment choices, especially for those that want to leverage existing or specialized component investments. In either case, enterprises can choose to implement the components of this tier to coexist with or access key aspects of Oracle Beehive.

Examples of supported Oracle and third-party components in the Ancillary Tier include:

- Oracle Internet Directory
- Oracle BPEL Process Manager

Note: Out of the box, Oracle Beehive comes pre-bundled with one Oracle BPEL Process Manager instance, although organizations may leverage other Oracle Beehive BPEL Process Manager instances if they exist.

- Oracle Secure Enterprise Search

- IBM Tivoli
- Microsoft Active Directory Server
- Microsoft Exchange Server
- Sun Java Directory Server
- Symantec Scan Engine

Connections Between Oracle Beehive Tiers

Components in the Oracle Beehive tiers connect through the following system frameworks and components:

- The Database Access Framework, which supports connections between the Application and Data Tiers. For more information, refer to "[Overview of the Database Access Framework](#)".
- The Beehive Transport Infrastructure (BTI), which supports client connectivity between the Client and Application Tiers. For more information, refer to "[Overview of the Beehive Transport Infrastructure](#)".
- The Event Framework, which manages all system events, as well as the workflows and notification logic that they initiate. For more information, refer to "[Overview of the Event Framework](#)".
- Oracle Beehive schemas, which segregate system, configuration, and collaborative data, and application code. For more information, refer to "[Overview of Oracle Beehive Schemas](#)".

Overview of the Database Access Framework

The Application and Data Tiers connect through the Database Access Framework, which provides a logical data structure and an optimized data management resource for Oracle Beehive. The Database Access Framework improves system performance and scaling capabilities, while reducing overall system complexity.

The Database Access Framework controls all access to the database through a connection pool that it manages. Services request connections to the database through the connection pool. Once a service receives the requested information, it returns the connection back to the connection pool. The Database Access Framework leverages Java Database Connectivity (JDBC) for these connections.

The Database Access Framework enables centralized management of data store resources and utilizations. It also simplifies database administration tasks that are external, yet related, to Oracle Beehive, including:

- Changing database connection information, such as database host names and Oracle RAC nodes
- Managing database connection pool sizes
- Gathering database usage metrics

Overview of the Beehive Transport Infrastructure

The Beehive Transport Infrastructure (BTI) enables connectivity between the Client and Application Tiers by providing the network-level infrastructure for both client-to-server and server-to-server communications. Specifically, the BTI provides port and connection management for Microsoft Outlook (with Oracle Beehive

Integration for Outlook) and non-HTTP, standard-based clients through its proprietary multiplexor protocol (MX).

In a typical enterprise client/server deployment, clients connect through a series of networks to a known port on a known server. However, in many cases, the communication channels must traverse firewalls, load balancing routers, forward and reverse proxies, and demilitarized zones (DMZs). The BTI is designed for deployment in these complex network infrastructures, while providing optimal port and connection management.

The BTI is a bundled component of the Oracle Beehive server so it is always provided whether you deploy Oracle Beehive in the Application Tier or in a DMZ. The latter case provides another layer of security between clients connecting to the system and internal components such as Oracle Database. This increased level of security and control over client connections is particularly beneficial when client connections are expected to originate from points outside of your network, such as from the Internet or from other private networks.

Oracle Beehive also provides Oracle HTTP Server, which enables client connectivity over HTTP/HTTPS. This can be leveraged in cases where HTTP/HTTPS is the preferred method of enabling client connections that originate from external locations.

Overview of the Event Framework

Oracle Beehive is an event-driven platform. Many aspects of system functionality and service interoperability stem from the event-based design of the platform. To manage this complex and powerful design implementation, Oracle Beehive provides the Event Framework. The Event Framework manages all events that occur in the system, which includes initiating any notification logic, workflows, and policies that may apply.

Oracle Beehive supports business events, which are events that trigger business logic, such as initiating workflows or applying policies. Business events provide a unique integration point for Oracle Beehive that allows customers to customize business processes or initiate external processes based on specific activities or occurrences. An example of a business event is when a user creates a document and it initiates a specific approval and notification process.

The open design of Oracle Beehive enables organizations to customize and extend the Event Framework by integrating workflows and policies. For example, developers can write and integrate custom business logic that executes when corresponding events are triggered in the system. Or, organizations can extend Oracle Beehive functionality by attaching custom programs to workflows that result from events. Oracle Beehive administrators can also customize Oracle Beehive business logic to suit their specific requirements.

Overview of Oracle Beehive Schemas

Oracle Beehive schemas logically segregate system, configuration, and collaborative data, and application code. In addition to organizing system, data, and configuration management, Oracle Beehive schemas also provide a significant security benefit by isolating each of the major data components that drive the system. Each Oracle Database in an Oracle Beehive deployment must contain one Oracle Beehive schema set.

An Oracle Beehive schema set consists of the following schema types:

- [Code Schema](#)
- [Data Schema](#)

Code Schema

The Code Schema contains all Procedural Language/Structured Query Language (PL/SQL) code used by Oracle Beehive. The Code Schema enables application-level cloning and code updates separate from system data. Benefits of these capabilities include easier and seamless patches and upgrades, since these tasks can be performed without any system downtime. The separation of PL/SQL code from system data also reduces the amount of time required for system backups, as code backups are required less frequently.

The Code Schema also controls the access requests made by Oracle Beehive services to data residing in the Data Tier, as well as the connections from services to other Oracle Beehive schemas, which the services cannot connect to directly. Connections between Oracle Beehive services and the Code Schema are managed by the Database Access Framework.

Data Schema

The Data Schema is the primary repository of collaborative data for Oracle Beehive. In Oracle Beehive, each service owns specific tables and views for the data that it manages. However, a service can request data from the tables or views owned by another service. For example, the Calendar Service may request data from a table belonging to the E-mail Service, which, in turn, may refer to and request information from tables owned by the User Directory Service.

Tools for Deploying Oracle Beehive

This module provides a high-level overview on the tools that can be used to deploy Oracle Beehive. This module contains the following topics:

- [The Oracle Beehive Install Wizard](#)
- [The Oracle Beehive Config Wizard](#)
- [Oracle Beehive Administration Console](#)
- [The Oracle Beehive beect1 Utility](#)
- [Oracle Enterprise Manager Grid Control](#)

The Oracle Beehive Install Wizard

The Oracle Beehive Install Wizard is a GUI-based wizard that guides you through the installation process. The Install Wizard is provided with your Oracle Beehive media and can be used to install, upgrade (Oracle Beehive Release 1.3 and later), and uninstall Oracle Beehive in either interactive mode or silent mode. In interactive mode, the Install Wizard prompts you for information and settings specific to your Oracle Beehive deployment and the supporting infrastructure. In silent mode, you supply a response file that contains all of the same information and settings normally requested during the Oracle Beehive installation process.

Note: Upgrades are supported for Oracle Beehive Release 1.2.1 or 1.2.2 to Oracle Beehive Release 1.3 only.

For more information on the Oracle Beehive Install Wizard, refer to the *Oracle Beehive Installation Guide* for your operating system.

The Oracle Beehive Config Wizard

The Oracle Beehive Config Wizard is a GUI-based wizard that guides you through the configuration segment of the installation process for software-only installations. Similar to the Oracle Beehive Install Wizard, it can be used in either interactive mode or silent mode. With the latter, you supply a response file that contains all of the same information and settings normally requested during the Oracle Beehive configuration process.

For more information on the Oracle Beehive Config Wizard, refer to the *Oracle Beehive Installation Guide* for your operating system.

Oracle Beehive Administration Console

Note: The Oracle Beehive Administration Console is provided in Oracle Beehive Release 1.3 and later.

The Oracle Beehive administration client, the Oracle Beehive Administration Console, is a secure, browser-based client built on Oracle ADF Faces technology. The Oracle Beehive Administration Console provides Oracle Beehive administrators centralized and role-based access to system configuration and management, user and group administration, monitoring, and reporting functions.

The Oracle Beehive Administration Console requires a server-based installation that is separate from the Oracle Beehive installation. The Oracle Beehive Administration Console installation program is provided with the Oracle Beehive installation kit. Once installed and configured, administrators simply need to access the configured URL to launch the Oracle Beehive Administration Console, log on to the system, and access the administrative features and information that it provides. The Oracle Beehive Administration Console supports the Hypertext Transfer Protocol Over Secure Socket Layer (HTTPS) for secure communications.

Note: For security reasons, Oracle strongly recommends enabling SSL encryption for your Oracle Beehive Administration Console installation.

For more information on how you can use the Oracle Beehive Administration Console to manage your Oracle Beehive deployment, please refer to *Oracle Beehive Administration Console Online Help*.

The Oracle Beehive `beectl` Utility

The Oracle Beehive `beectl` utility is a command-line tool provided with Oracle Beehive. Administrators can use `beectl` to manage their Oracle Beehive deployments by executing commands in a supported operating system shell. Some of the deployment-related tasks for which administrators can use the utility include:

- Cloning Oracle Beehive instances and OC4Js
- Creating and managing client application configuration files
- Provisioning client applications

For more information on how you can use `beectl` to manage your Oracle Beehive deployment, refer to the *Oracle Beehive Administrator's Guide*.

Oracle Enterprise Manager Grid Control

Oracle Enterprise Manager Grid Control includes a Web-based administration tool that provides both installation and monitoring capabilities for Oracle Beehive. 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. It provides a broad set of administration, configuration management, provisioning, end-to-end monitoring, and security capabilities. Due to these capabilities and others, Oracle Enterprise Manager Grid Control is

recommended for use with Oracle Beehive. Oracle Beehive supports Oracle Enterprise Manager 10g Release 4 Grid Control (10.2.0.4) or later.

Installing Oracle Beehive through Oracle Enterprise Manager Grid Control requires the Oracle Beehive Provisioning Application, which is provided with Oracle Beehive. However, Oracle Enterprise Manager Grid Control is not provided with the Oracle Beehive media package and must be installed separately. For details, please contact your Oracle representative.

For more information on how you can use Oracle Enterprise Manager Grid Control to install Oracle Beehive, refer to the *Oracle Beehive Installation Guide* for your operating system. For more information on how you can use Oracle Enterprise Manager Grid Control to manage your Oracle Beehive deployment, refer to the *Oracle Beehive Administrator's Guide* or your Oracle Enterprise Manager Grid Control documentation.

Deploying Oracle Beehive with Other Software Components

This module provides overviews of the various Oracle and third-party components supported by Oracle Beehive and the related considerations for deploying them with the system.

This module contains the following topics:

- [Deploying Oracle Beehive with Oracle Database](#)
- [Deploying Oracle Beehive with Oracle Real Application Clusters \(Oracle RAC\)](#)
- [Deploying Oracle Beehive with an External User Directory](#)
- [Deploying Oracle Beehive with Oracle Universal Records Management \(URM\)](#)
- [Deploying Oracle Beehive with Oracle Wallet](#)
- [Deploying Oracle Beehive with an External Oracle BPEL Process Manager Instance](#)
- [Deploying Oracle Beehive with Oracle Secure Enterprise Search 10g](#)
- [Deploying Oracle Beehive with Microsoft Exchange Server 2003](#)
- [Deploying Oracle Beehive with Symantec Scan Engine](#)

Deploying Oracle Beehive with Oracle Database

Oracle Beehive stores all of its data—collaborative, configuration, and audit data, as well as its log archives—in Oracle Database. Every Oracle Beehive deployment must have at least one Oracle Database instance installed and available prior to installing Oracle Beehive. Oracle Beehive supports Oracle Database 10g Release 2 (10.2.0.3) Patch Set 2 and Oracle Database 11g Release 1 (11.1.0.6).

Prior to installing Oracle Beehive and deploying it with Oracle Database, there are several required steps that you must take and a few recommended configuration steps you should consider. These steps will prepare your database environment for certain Oracle Beehive installation prerequisites and other requirements, and include:

- Installing any patches required for your version of Oracle Database.
- Setting the minimum values for certain Oracle Database initialization parameters, including `java_pool_size`, `job_queue_processes`, and `processes`.
- Enabling the system's archiving capabilities.
- If you plan to use a database that uses raw storage, customizing Oracle Beehive's custom tablespaces script (`beehive_custom_ts.sql`).

For more information on these and other database-related steps, including details on how to perform these steps, please refer to the *Oracle Beehive Installation Guide* for your operating system.

Notes:

- For high availability deployments with a shared file system (or that leverage the `filesystem_reference` object within workspaces), all computers on which Oracle Beehive Application Tier instances and Oracle Database instances reside must have access to the file system reference paths at the same logical location. This shared access may be accomplished using a Network File System (NFS) server, symbolic links (symlinks), or another supported method. Typically, organizations will experience optimal performance if their file systems reside on computers other than those on which Oracle Beehive and Oracle Database reside.
 - Oracle Beehive supports the option to deploy a secondary database instance dedicated to the system's search functions. This option should be considered in large deployments as it may provide significant performance improvements for search-related features. For more information on this option, please contact your Oracle Support representative.
-
-

Deploying Oracle Beehive with Oracle Real Application Clusters (Oracle RAC)

Oracle Beehive supports Oracle Real Application Clusters (Oracle RAC). With Oracle RAC, you can deploy Oracle Database across multiple computers so that they each share a single physical database. This is useful for deployments that need to achieve high availability and scalability, especially on low cost hardware. With Oracle RAC, the number of computers across which Oracle Database is distributed is invisible to Oracle Beehive and its users.

If you plan to deploy Oracle Beehive with Oracle RAC, you should consider the pre- and post-installation steps necessary to prepare your deployment for this option. These topics are covered in the following sections:

- [Preparing to Deploy Oracle Beehive with Oracle RAC](#)
- [Post-installation Steps for Deploying Oracle Beehive with Oracle RAC](#)

Preparing to Deploy Oracle Beehive with Oracle RAC

Prior to installing Oracle Beehive and deploying it with Oracle RAC, there are several required configuration steps that you must take and a few recommended steps that you should consider. These steps, which will prepare your high availability environment for certain Oracle Beehive installation prerequisites and other requirements, include:

- Creating multiple logical database services.
- Starting all of the logical database services that you created.
- Enabling X/Open Distributed Transaction Processing (DTP) for all database services.

- For deployments that require communications over Secure Sockets Layer (SSL), enabling Secure Oracle Notification Services (ONS) notification.
- For databases that use raw storage, creating the `ORABPEL` tablespace manually.

For more information on these and other pre-installation steps related to Oracle RAC, including details on how to perform these steps, please refer to the *Oracle Beehive Installation Guide* for your operating system.

Post-installation Steps for Deploying Oracle Beehive with Oracle RAC

After installing Oracle Beehive, you must complete several post-installation steps, including:

- Retrieving your site and database object system identifiers (you'll need these for the post-installation steps that follow).
- Changing the value of the `connect_string` property by substituting the database service name with the global service name.
- Updating the database system object and the local bootstrap file with the updated connect string value.
- Modifying the database system object's configuration by specifying the database services you created and the ONS remote ports that are listening on your Oracle RAC nodes.
- Committing the changes listed above to all Oracle Beehive instances.
- Configuring your Oracle RAC nodes to receive secure ONS notifications (if they haven't been configured already).

For more information on these and other post-installation steps related to Oracle RAC, including details on how to perform these steps, please refer to the *Oracle Beehive Installation Guide* for your operating system.

Deploying Oracle Beehive with an External User Directory

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 (UDS).

Currently, Oracle Beehive supports the following user directory servers:

- Oracle Internet Directory
- IBM Tivoli
- Microsoft Active Directory Server
- Sun Java System Directory Server

Organizations can also leverage external user directories to manage user authentication attributes such as usernames and passwords. This option requires Oracle Beehive administrators to configure the Authentication Service after installation using the `beectl` command-line utility. For more information regarding this option, please refer to the *Oracle Beehive Administrator's Guide*.

If you choose to integrate Oracle Beehive with a supported external user directory, you should consider the limitations of this option as well as the steps to prepare your deployment for this option. These topics are covered in the following sections:

- [Preparing to Deploy Oracle Beehive with an External User Directory](#)
- [Limitations of Deploying Oracle Beehive with an External User Directory](#)

Preparing to Deploy Oracle Beehive with an External User Directory

After installing Oracle Beehive but before deploying the system with an external user directory, you must import user account data from your external user directory server instance to UDS. This process involves creating the following XML files:

- **LDAP mapping profile:** An XML file that contains external user directory server settings and specifies how to convert those entries to Oracle Beehive users and groups. This involves specifying attribute mappings between those defined in an external directory server and those used by Oracle Beehive.
- **User file:** An XML file that represents, in a format specified by Oracle Beehive, all the user accounts that need to be synchronized. To create this, administrators use the LDAP mapping profile and the `beectl download_external_user_data` command.

For more information, including the steps to create these files, please refer to the *Oracle Beehive Installation Guide* for your operating system.

Limitations of Deploying Oracle Beehive with an External User Directory

Despite the multiple options and benefits provided by Oracle Beehive's support of external user directories, certain limitations exist and include:

- Oracle Beehive does not support integration with multiple, unique user directories. Oracle Beehive can only integrate with a single user directory instance on a specific server.
- The user directory option in which you initially deploy Oracle Beehive is fixed. You cannot migrate to another supported option after the system is deployed. For example, if you choose to initially deploy Oracle Beehive leveraging its native user directory, you cannot change to an external user directory later. Therefore, Oracle strongly recommends that you consider and choose your user directory option *before* deploying Oracle Beehive.
- User accounts can only be mastered in one place, either natively in Oracle Beehive or in an external user directory. Despite this requirement, administrators may choose to master most user account attributes in one directory even if the user accounts themselves are mastered in another directory. However, certain user account attributes that are specific to Oracle Beehive, such as voicemail passwords and instant messaging user names, can only be mastered in Oracle Beehive.
- To manage user account attributes that are mastered in external LDAP directories, administrators can only use the tools provided with or for those directories. Administrators cannot use the tools provided with Oracle Beehive, such as `beectl`, to manage user account attributes that are mastered in external LDAP directories.
- The synchronization process between external user directory servers and Oracle Beehive is unidirectional, that is, changes in an external user directory are imported into Oracle Beehive only. Oracle Beehive cannot promote the user account attributes that it manages to external user directories.

Deploying Oracle Beehive with Oracle Universal Records Management (URM)

Oracle Universal Records Management (Oracle URM) enables organizations to manage their records and retention policies, disposition processes, and litigation holds or freezes in a central repository known as a Universal Records Management (URM) server. Organizations can then apply those policies, dispositions, and holds to content stored in other systems, such as Oracle Beehive. Oracle Beehive provides integration with Oracle URM through the Records Management Service.

Deploying Oracle Beehive with Oracle URM requires that you have an Oracle URM instance installed and configured prior to installing Oracle Beehive. After you install Oracle Beehive, you must complete several tasks, some of which you perform in Oracle Beehive and others in Oracle URM. These tasks are grouped as follows:

- [Oracle Beehive-based Tasks for Deploying Oracle Beehive with Oracle URM](#)
- [Oracle URM-based Tasks for Deploying Oracle Beehive with Oracle URM](#)

Oracle Beehive-based Tasks for Deploying Oracle Beehive with Oracle URM

Deploying Oracle Beehive with Oracle URM requires completion of the following tasks in Oracle Beehive:

1. Specifying (registering) the Oracle URM instance that you want Oracle Beehive to leverage. To do this, you issue the `add_urm` and `activate_configuration` commands through the `beectl` utility.
2. Starting the Records Management Service through `opmnctl`.

For more information on these tasks, please refer to the *Oracle Beehive Administrator's Guide*.

Oracle URM-based Tasks for Deploying Oracle Beehive with Oracle URM

Deploying Oracle Beehive with Oracle URM requires logging in to Oracle URM as a user with administrator privileges and completing the following tasks:

- Verifying that the `sysadmin` user has been granted all possible roles in Oracle URM. This includes the `rma`, `rmaadmin`, `admin`, `ermadmin`, `ermrequestor`, `rmaprivileged`, and `sysmanager` roles.
- Creating retention categories and folders that are required by your organization for content managed by Oracle Beehive. You can create retention categories within record series or file plans. You can create retention folders within retention categories and other retention folders. Once created, you can view retention categories and folders in Oracle Beehive by issuing the `list_file_plan` command through `beectl`.
- Creating disposition rules required by your organization for content managed by Oracle Beehive.

Note: Currently Oracle Beehive only supports the `destroy` disposition action. Oracle Beehive will not process any other disposition actions that are defined and assigned to retention categories.

- Enabling the dispositions for the content managed by your Oracle Beehive instance.

For more information on these tasks, please refer to the documentation provided with your Oracle URM instance.

Deploying Oracle Beehive with Oracle Wallet

Oracle Wallet is a component of Oracle Application Server 10g that provides important authentication capabilities. 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 Secure Sockets Layer (SSL) for strong authentication. Oracle Wallet provides an encrypted Transport Layer Security (TLS) communication channel that some Oracle Beehive services require, such as the XMPP Service and the Workflow Service. Oracle Wallet is also required when configuring Oracle Beehive Web Services for Security Assertions Markup Language (SAML) authentication.

After installing Oracle Beehive, you should configure Oracle Beehive to use Oracle Wallet so that clients may access the system with a TLS connection. These steps include:

- Enabling auto-login for Oracle Wallet.
- Enabling Oracle Remote Method Invocation over SSL (ORMIS) for Oracle Wallet.
- Configuring your Oracle Beehive server instance to use Oracle Wallet. If your deployment contains multiple instances, you will need to configure each instance separately.

For more information on the specific steps required to deploy Oracle Beehive with Oracle Wallet, refer to the *Oracle Beehive Installation Guide* for your operating system.

Deploying Oracle Beehive with an External Oracle BPEL Process Manager Instance

Out of the box, Oracle Beehive comes pre-bundled with one Oracle BPEL Process Manager instance that is configured with a few default workflows through the Workflow Service. You may leverage this internal instance or, if your organization has an existing Oracle Beehive BPEL Process Manager instance, you may integrate Oracle Beehive with that one instead. This section discusses the considerations related to the integrating Oracle Beehive with an existing Oracle Beehive BPEL Process Manager instance.

Caution: If you plan to deploy Oracle Beehive with an external BPEL Process Manager instance, Oracle strongly recommends that you configure this integration immediately after you install Oracle Beehive, or very soon thereafter. You may encounter issues if you initially deploy Oracle Beehive with its internal BPEL Process Manager instance and later switch to an external BPEL Process Manager instance.

To deploy Oracle Beehive with an external Oracle BPEL Process Manager instance, you must complete the following tasks after you install Oracle Beehive:

- Specify the external Oracle BPEL Process Manager instance and BPEL cluster that you want Oracle Beehive to use. You use the `beectl` utility to complete these tasks.
- In the ORABPEL repository for your external instance, create a synonym to the Oracle Beehive Workflow PL/SQL package. Additionally, you may create a database link to this package if the schemas for the external ORABPEL repository and Oracle Beehive are not stored in the same database.
- Deploy the Oracle Beehive Identity Service Provider package (`isprovider.jar`) and configuration file (`is_config.xml`) to your external Oracle BPEL Process Manager instance. You complete this task either through auto-deploy mode of Oracle BPEL Process Manager or by using the Oracle BPEL Console.

Note: After you deploy the Oracle Beehive Identity Service Provider package, any existing Identity Service configurations for your external Oracle BPEL Process Manager instance will no longer work. However, you may optionally merge the Oracle Beehive `is_config.xml` file with the one in your external Oracle BPEL Process Manager instance. This will allow the Oracle Beehive Identity Service provider and other custom providers to coexist. Furthermore, if you integrate Oracle Beehive with an external user directory (LDAP) that is already configured for your external Oracle BPEL Process Manager instance, then you do not have to complete this step. For more information on this option, please refer to the *Oracle BPEL Process Manager Administrator's Guide*.

- Deploy the default Oracle Beehive BPEL workflows, as well as any custom workflows, to your external Oracle BPEL Process Manager instance. To deploy the default workflows, you use the `beectl` utility. To deploy custom workflows, you also use the `beectl` utility (to register the workflows) and either auto-deploy mode of Oracle BPEL Process Manager or the Oracle BPEL Console (to manually deploy the custom workflow suitcase into your external instance).

Notes:

- For more information on the specific steps required to deploy Oracle Beehive with an external Oracle BPEL Process Manager instance, please refer to the *Oracle Beehive Installation Guide* for your operating system.
 - For more information on the tasks that are related to the pre-bundled internal Oracle BPEL Process Manager and its workflows, please refer to "Managing Oracle Beehive Events, Policies, and Workflows" in the *Oracle Beehive Administrator's Guide*.
 - For more information on considerations and tasks that are specific to Oracle BPEL Process Manager, and especially external instances of it, please refer to the *Oracle BPEL Process Manager Administrator's Guide*.
-
-

Deploying Oracle Beehive with Oracle Secure Enterprise Search 10g

Oracle Beehive maintains its own optimized search index enabling users to perform comprehensive searches across all Oracle Beehive artifacts. At the enterprise level, however, other information repositories might exist and contain information that users need. For example, depending on their roles, knowledge workers might need to find expense reports or purchase requisitions stored outside of Oracle Beehive. This level of search across all enterprise information repositories is provided by Oracle Secure Enterprise Search.

Oracle Secure Enterprise Search has been designed as 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.

To leverage this powerful option requires the completion of tasks (post-installation) that are specific to Oracle Beehive and others that are specific to Oracle Secure Enterprise Search 10g. For example, your Oracle Beehive instance must be configured in Oracle Secure Enterprise Search 10g as a federated data source.

You use the `beectl` utility to complete the tasks that are specific to Oracle Beehive, which include:

- Creating a special user account that has administrator rights to the content managed by Oracle Beehive.
- Configuring the Secure Enterprise Search service by entering the host and port number of your Oracle Beehive Application Tier instance, and by enabling the service.
- Activating these changes using the `activate_configuration` command.

You use the Oracle Secure Enterprise Search Administration Tool to complete the tasks that are specific to Oracle Secure Enterprise Search 10g, which include:

- Activating the identity plug-in on the Global Settings - Identity Management Setup page.
- Specifying your Oracle Beehive database instance as a federated source.
- Specifying the Uniform Resource Identifier (URI) of the Oracle Beehive Search Service.
- Entering the credentials of the special user account that you created in Oracle Beehive.

For more information on the tasks that are specific to Oracle Beehive, please refer to the *Oracle Beehive Installation Guide* for your operating system. For more information on the tasks that are specific to Oracle Secure Enterprise Search 10g, please refer to "Setting up Federated Sources" in the *Oracle Secure Enterprise Search 10g Administrator's Guide*.

Deploying Oracle Beehive with Microsoft Exchange Server 2003

Oracle Beehive supports integration with Microsoft Exchange Server 2003 through the Oracle Collaboration Coexistence Gateway. The Oracle Collaboration Coexistence Gateway is an Oracle proprietary solution that allows Oracle Beehive users to collaborate with Microsoft Exchange users beyond the limited capabilities of e-mail. Additionally, this solution allows Microsoft Exchange users to leverage Oracle Beehive features without being migrated from Microsoft Exchange.

To learn more about how to deploy Oracle Beehive with Microsoft Exchange Server 2003, refer to the following topics:

- [Installing and Configuring the Oracle Collaboration Coexistence Gateway](#)
- [Load Balancer Requirements for Deploying Oracle Beehive with Microsoft Exchange Server 2003](#)

Installing and Configuring the Oracle Collaboration Coexistence Gateway

To integrate Oracle Beehive with Microsoft Exchange Server 2003, you install and configure the following Oracle Collaboration Coexistence Gateway components on one or more computers in your Microsoft Exchange domain:

- Oracle Coexistence Connector for Microsoft Exchange Server
- Oracle Change Notification Service for Microsoft Exchange Server

You install both of these component separately on one or more computers in your Microsoft Exchange domain using the Oracle Beehive Install Wizard. Although you may install both components on the same computer, it is not recommended that you do so. Instead, it is recommended that you install these components on separate computers that each host their own Microsoft Exchange Server 2003 instances. Furthermore, you should install the Oracle Change Notification Service only on computers that host Microsoft Exchange user mailboxes and the Oracle Coexistence Connector for Microsoft Exchange Server only on computers that do not host user mailboxes. You may also install these components on computers in one or more Microsoft Exchange routing groups.

After installing Oracle Collaboration Coexistence Gateway components, you must perform several post-installation tasks, which include configuring the Coexistence Service and configuring your deployment's load balancers (if required). You perform these tasks manually outside of the Oracle Beehive Install Wizard.

Notes:

- For more information on specific prerequisites and steps related to installing the Oracle Collaboration Coexistence Gateway and its components, please refer to "Oracle Collaboration Coexistence Gateway Install Help" in the *Oracle Beehive Installation Guide for Microsoft Windows*.
 - For information on configuring the Coexistence Service, please refer to the *Oracle Beehive Administrator's Guide*.
 - For information on any load balancer requirements that might apply, please refer to "[Load Balancer Requirements for Deploying Oracle Beehive with Microsoft Exchange Server 2003](#)".
-
-

Load Balancer Requirements for Deploying Oracle Beehive with Microsoft Exchange Server 2003

Depending on the message type, the Oracle Collaboration Coexistence Gateway sends messages between Oracle Beehive and Microsoft Exchange Server 2003 over HTTP/HTTPS or SMTP. Therefore, you will need one or more load balancers if your integration plans include more than one Oracle Beehive Application Tier instance or more than one Oracle Coexistence Connector instance. In these cases, load balancers are required to manage the HTTP/HTTPS traffic and, potentially, the SMTP traffic between Oracle Beehive and Microsoft Exchange Server. This requirement is due to the fact that the Coexistence Service (which resides on all Oracle Beehive server instances) and the Oracle Coexistence Connector (which must reside on at least one Microsoft

Exchange Server instance) accept only a single URL representing their connection endpoints. Although it is not required to configure load balancers for SMTP-based coexistence traffic, it is recommended that you do so.

Deploying Oracle Beehive with Symantec Scan Engine

Oracle Beehive supports integration with Symantec Scan Engine. This provides another option for organizations that want to leverage existing Symantec Scan Engine instances or that want anti-virus features beyond what the Oracle Beehive Virus Scanner provides. Through this integration, organizations can leverage the scan types and modes that Symantec Scan Engine provides, as well as its artifact and message filtering capabilities. Oracle Beehive supports Symantec Scan Engine version 5.1.2 or later.

Administrators manage the Oracle Beehive Virus Scanner through `beectl`. To manage Symantec Scan Engine, administrators should leverage the tools provided with that product.

For details on integrating Symantec Scan Engine with Oracle Beehive, refer to "Adding a Virus Engine to Oracle Beehive" in the *Oracle Beehive Administrator's Guide*. To configure Symantec Scan Engine to scan e-mail messages, refer to "Managing Attachment Blocking and Virus Scanning" in the *Oracle Beehive Administrator's Guide*.

Network Considerations for Oracle Beehive

This module discusses the following network considerations when planning for an Oracle Beehive deployment:

- [Basic Network Considerations](#)
- [Network Considerations for Small Deployments](#)
- [Access and Security Considerations](#)
- [Traffic Considerations](#)
- [Other Network Considerations for Deploying Oracle Beehive](#)

Note: Because network configuration varies according to the unique requirements of each organization, this module does not provide specific network setup instructions. Rather, it discusses the general network considerations that organizations must address when deploying Oracle Beehive.

Basic Network Considerations

The basic goal of network planning is to make optimal use of Oracle Beehive and your existing network services and components. To accomplish this, you must determine a number of factors including the amount of access permitted to external sources, the capacity needed to meet user requirements, and network availability. You must also consider business factors such as budget constraints and the strategic reasons behind each of your network planning goals.

Network Considerations for Small Deployments

In small deployments of Oracle Beehive, network and architectural planning are intertwined since smaller pools of available servers result in fewer deployment options. When planning an architecture with only one or two servers, you will often have to choose between component accessibility and server location. The location of servers is often based on security requirements and, as a result, access to the servers must be planned accordingly.

Access and Security Considerations

When deploying Oracle Beehive, access and security must be planned simultaneously since each aspect has a direct impact on the other, especially with smaller deployments.

Access is arguably the more complex of these two considerations. In virtually any deployment, you must account for a number of access methods (or levels) within a single environment. These depend on factors such as the locations from which users will access the network, the types of clients users will leverage, and the components to which users will need access.

Because access is a network-dependent element, much of its setup depends on the surrounding network environment. Therefore, Domain Name Services (DNS), mail relays, reverse proxies, firewalls, network routing, and other network services must interact very closely with the components of Oracle Beehive. Understanding this interaction is the key to understanding Oracle Beehive network access in a production environment.

Security is a subset of access in many ways, as many security issues are solved simply by denying access to certain services or networks. For example, an organization might choose to deny access from outside its network across all protocols except for HyperText Transfer Protocol (HTTP) and Secure HyperText Transfer Protocol (HTTPS). HTTPS is the HTTP protocol implemented over Secure Sockets Layer (SSL) or Transport Layer Security (TLS). It is more secure than HTTP because it provides authentication and data encryption. Users typically access Oracle Beehive from outside the network with supported clients over HTTPS. So, you can proxy HTTPS as each user requires.

The other element to consider is how to secure the traffic that you allow to flow, and this is where encryption plays an important role. This is especially true for external traffic.

Traffic Considerations

A typical Oracle Beehive deployment may separate traffic in the following ways:

- [Internal Traffic](#)
- [External Traffic](#)

Internal Traffic

Internally, all protocols are allowed and all clients may be used. In this case, a typical deployment could leverage the BTI for all client connections. This policy can be modified to be more granular, either by configuring access rules or by separating the networks with firewalls.

External Traffic

Externally, only a few encrypted, hardened, and monitored services are exposed. For example, some organizations choose to make external client access available only over HTTP and HTTPS. In this case, a typical deployment could leverage Oracle HTTP Server (installed with Oracle Beehive) deployed in a DMZ.

Other Network Considerations for Deploying Oracle Beehive

This section provides a list of questions that you should ask before deploying Oracle Beehive. The answers to these questions will help you to evaluate which Oracle Beehive and network components need to be deployed, where they need to be deployed, and how they need to be configured:

- [What kind of network and security policies does your organization have?](#)

- Does your organization have a DMZ and if so, what kind?
- Does your organization use NAT to abstract its network?
- Has your organization implemented load balancing and reverse proxy?
- Does your organization use hardware encryption acceleration?
- Does your organization have an SMTP mail relay?
- Does your organization want to expose IMAP externally?
- Is filtering installed for spam and viruses?
- What components might your organization want to deploy initially or later?
- Which Oracle Beehive services and network services will be used?
- From where should Oracle Beehive be accessible?
- Do your plans fit into your organization's security model?
- What are the existing network services on which you will rely?
- What is the configurability of your network services? Can they be easily changed to suit Oracle Beehive?
- What is the size of the current network?
- How many users will there be?
- What is your budget?

What kind of network and security policies does your organization have?

Unless your organization is willing to revise its policies, you'll need to adhere to existing network and security policies when deploying Oracle Beehive. This is likely to affect where you deploy Oracle Beehive, such as in or behind a DMZ, and what components you'll use to enable client connections, such as Oracle HTTP Server or the BTI.

Does your organization have a DMZ and if so, what kind?

This is crucial to the design and planning of your architecture, as it determines where components need to be placed, depending on how they are to be accessed. DMZ is a part of the network, which is in between an intranet and the Internet, and is often referred as the neutral zone. It enables only certain services of the hosts in an intranet to be accessible to the hosts on the Internet. This sub-network is especially used for public access servers such as Web servers.

If there is a DMZ:

- How is it used for external access?
- How is it used for internal access?
- Are DNS servers used? If so, consider the following issues.
 - Sometimes, two DNS servers (split DNS) are used to help separate deployments by reserving server names for separate internal and external computers.
 - DNS servers can be separated onto separate networks; for example, one in the DMZ and one in the internal network.
 - You can configure any DNS server, such as Berkeley Internet Name Domain (BIND), to respond to DNS requests differently, depending on where the request originates.

- DNS services can be further fortified when combined with Network Address Translation (NAT).

Using two DNS servers provides more flexibility than using just one. In either case, clients connect to either an external or internal IP address for the same host name, depending on where the connection originates, and client access is properly routed by DNS connectivity.

Does your organization use NAT to abstract its network?

NAT is the process of translating an IP address used within one network to a different IP address of another network. Generally, organizations set up NAT to translate internal addresses into public routable addresses. The translation takes place at the firewall and protects internal addresses from external tracing or routing.

NAT has a large and beneficial impact on naming services. External IP addresses in DNS can be mapped to virtual addresses as configured in Oracle Beehive.

Note: When you install Oracle Beehive, each Oracle Beehive server instance assumes the name of the server on which it is installed. You must configure external IPs on your DNS server, then configure Oracle Beehive to use the appropriate virtual addresses instead of the original server name.

Has your organization implemented load balancing and reverse proxy?

Load balancing is the process of managing the distribution of inbound traffic among multiple servers. The use of load balancers and reverse proxy is an important strategy in securing your Oracle Beehive deployment. With load balancing and reverse proxy, you can abstract the routing and addressing of HTTP/HTTPS traffic so that Oracle Beehive servers are not directly exposed to external users.

If your organization uses existing hardware and a switching strategy, consider whether Oracle Beehive can be integrated into it.

Using content switches as load balancers is recommended. Content switches can host virtual IP addresses and spread the load over multiple servers, particularly for Application Tier services. Traffic can be redirected based on protocols such as HTTP, HTTPS, SMTP, and so forth.

If your organization has host names for its current services running on virtual devices, migration will be much easier. In fact, it is always a good idea to identify virtual services names early in the process. Embed them in documentation and get SSL certificates early if they are needed. It is also a good idea to start setting up DNS and Mail Exchange records early on, to map domain names to mail servers.

Does your organization use hardware encryption acceleration?

If you plan to use the SSL protocol to encrypt communications, it is a good idea to make use of SSL accelerators to optimize performance. SSL processing is a CPU-intensive task that can reduce performance. This is best off-loaded to hardware devices designed for this purpose.

Consider using SSL to encrypt external access to Oracle Beehive.

Does your organization have an SMTP mail relay?

Relay services are important for filtering, limiting, and routing SMTP traffic. They can be used to protect Oracle Beehive instances from external abuse and unnecessary load.

Most organizations have an existing SMTP mail relay (or Mail Transport Agent), ideally with spam services implemented on it. However, note the limitations of mail relays, such as:

- Attachment size
- Throughput throttling
- Retry count

Evaluate whether or not these limitations are realistic for your organization, and keep them in mind as you deploy Oracle Beehive.

It is a good strategy for many deployments to use programs such as these to handle and filter e-mail before forwarding it to the Oracle Beehive mail relay, which can reside behind a firewall or in a DMZ.

Does your organization want to expose IMAP externally?

IMAP is used to access e-mail messages from the mail server. Using IMAP, you can retrieve e-mail messages and manipulate them on the server itself. Many organizations choose not to provide IMAP access directly over the Internet, as they feel it overly exposes them from a security standpoint. If you decide to provide IMAP access across the Internet (without the additional security of a VPN infrastructure, for instance), Oracle Corporation strongly recommends implementing SSL for those IMAP services.

Is filtering installed for spam and viruses?

For the judicious use of valuable database storage, avoid saving unnecessary information such as e-mail spam. To protect the integrity of the network, viruses (often linked directly to spam) must also be blocked. There are a number of methods of filtering out these unwanted communications.

What components might your organization want to deploy initially or later?

It is possible that in the process of deploying Oracle Beehive, your organization may want to either implement components for the first time (such as a load balancer, for example) or upgrade existing ones (such as a mail relay).

Which Oracle Beehive services and network services will be used?

Determine which Oracle Beehive services your organization will use and consider which network services are necessary to support them. For example, if your organization plans to leverage the E-mail Service, you may want it to be accessible from anywhere in the world through HTTPS. Generally, the most important Oracle Beehive network components are as follows:

- **Routing Equipment:** Routing equipment can include routers, load balancers, and firewalls.
- **Domain Name System (DNS):** This network service resolves domain names or host names into Internet Protocol (IP) addresses. DNS maintains a database of all the host names and their respective IP address information.
- **Mail relay:** This network service is a server that routes e-mail messages to their destinations, mostly within local networks.

From where should Oracle Beehive be accessible?

For example, will it be accessible to users externally, internally, or both?

Do your plans fit into your organization's security model?

Before creating an elaborate deployment plan for Oracle Beehive, make sure you know your organization's security model, if there is one. For instance, some organizations may not allow DMZs or external access to e-mail.

What are the existing network services on which you will rely?

It may be practical to make use of existing network services.

What is the configurability of your network services? Can they be easily changed to suit Oracle Beehive?

Check how flexible your existing network implementation is.

What is the size of the current network?

Find out what sort of upgrades and additions, if any, will have to be made to the network to accommodate Oracle Beehive.

How many users will there be?

This is a primary concern in determining the type of deployment you need but it also affects (or may be affected by) the capacity of your network. If supporting a specified number of users is the highest priority, you'll need to determine whether or not your network has the speed, bandwidth, and latency to support those users, and what network improvements might be needed to support them. Or, if your network is a fixed and known quantity that cannot be changed, you'll need to determine how many users it can realistically support and then plan accordingly.

What is your budget?

Cost is always an important consideration in any network installation.

Architectural Considerations for Oracle Beehive

This module includes information on the following topics, which are related to the architectural considerations when planning for an Oracle Beehive deployment:

- [Basic Architectural Considerations](#)
- [Access Considerations](#)
- [Capacity Considerations](#)
- [Recovery Considerations](#)
- [Availability Considerations](#)
- [Scalability Considerations](#)
- [Workload Considerations](#)
- [Security Considerations](#)
- [Organization-specific Architecture Considerations](#)

Basic Architectural Considerations

Oracle Beehive can work in many different permutations, with exact placement and integration of components and servers varying greatly. Configuration depends on the needs of the organization and the existing architecture that can be leveraged.

The choice of architecture has a definite impact on performance and stability. Some organizations may require high availability, and may have moved to Oracle Beehive for this purpose. Others may not need high availability throughout an architecture, and can save time and resources initially.

Access Considerations

Access to Oracle Beehive is determined by your network setup and security strategies. Part of your planning involves determining where components need to be accessed from, whether internally, externally, or both. You can deploy multiple Oracle Beehive server instances in the Application Tier, depending on where your users are, perhaps with one in a DMZ and another on the internal network.

Capacity Considerations

To provide the appropriate capacity for the number of users in your deployment, you must start by evaluating how many servers you will need. Keep in mind not just the number of users, but also the frequency and concurrency of use.

Properly deployed Application Tier components can back up each other in cases of failures among them. This style of duplication and redundancy provides deployment flexibility and enables the load to be spread across all the components of the Application Tier.

Recovery Considerations

Make sure you plan for the ability to restore deleted or corrupt data to its last known good state. This is a part of your backup strategy.

Availability Considerations

Availability encompasses strategies you put into place to keep things running in case of system failures. Strategies can range from backing up data in-house to maintaining high levels of redundancy for Oracle Beehive system components. One example is to ensure you have network redundancy, so that in the event that one of your network paths fails, network traffic can be routed along a secondary or backup network route.

Scalability Considerations

With any deployment, it is important to plan for growth. You must be prepared to expand your deployment along with the size or needs of your user base. The use of virtual host names is recommended, even if you are installing on a single host or in a non-high availability environment. This will allow you to abstract Oracle Beehive server instances across multiple servers later on, without having to rename servers.

Make sure you have hardware that allows for growth and memory expansion. See the ["Capacity Considerations"](#) section for more information.

Workload Considerations

You can deploy Oracle Beehive on a single computer. However, such a deployment architecture may not be able to meet all the availability, security, and scalability requirements of your organization. Distribution of Oracle Beehive on multiple computers makes it easier to meet these requirements.

To optimize resource utilization on a particular computer, you should not mix different types of workloads on the computer. Typically, architectures for relatively large systems feature the distribution of these tiers on different computers.

The Application Tier, where Oracle Beehive server instances reside, and the Data Tier, where Oracle Database resides, handle different types of workloads. The Application Tier is CPU- and memory-intensive with fewer disk input/output (I/O) operations. Conversely, the Data Tier is I/O intensive.

In Oracle Beehive, the demand for Application Tier resources increases at a rate that is different from the rate at which the demand for Data Tier resources increases. This is another factor that justifies a distributed architecture. With a distributed architecture, you can scale up the Application and Data Tiers independently.

Security Considerations

If you want to provide access to Oracle Beehive components from public networks, you must take measures to secure this type of access. You can enhance the security of the system by distributing the tiers and deploying the system behind a DMZ. Standard security practices and Oracle Beehive prevent, for example, network traffic from passing directly from the Internet to the database. Instead, such traffic would be routed through a DMZ and a server in the Application Tier.

If you choose to deploy Oracle Beehive in a DMZ, the DMZ must contain the hardware and software required to relay traffic securely between the public network and private network.

Organization-specific Architecture Considerations

This section discusses deployment configuration considerations that may vary according to the specific requirements of each organization. For this reason, this and other Oracle Beehive deployment modules provide general, and not specific, guidance for these aspects of the system:

- [How many servers is your organization willing to manage?](#)
- [What type of platform is your organization comfortable with?](#)
- [Is your organization prepared to deploy Oracle Beehive in stages?](#)
- [Does your organization need high availability?](#)
- [What is your organization's existing backup solution?](#)

How many servers is your organization willing to manage?

This largely depends on the size of your IT staff and availability of system management tools.

What type of platform is your organization comfortable with?

Oracle Beehive supports several chip architectures and operating systems including heterogeneous and homogeneous platforms.

Is your organization prepared to deploy Oracle Beehive in stages?

It is strongly recommended that you consider deploying Oracle Beehive in stages, as described in the section "[Deploying Oracle Beehive in Stages](#)" from [Chapter 1, "Overview of Deploying Oracle Beehive"](#). At minimum, you should plan to deploy the system in the following stages:

- **Development:** Map out host names, servers, components, services, and access on a small scale.
- **Staging:** Expand your initial deployment for use by a test group, to expose configuration issue flaws as well as to test capacity and performance.
- **Production:** Official live deployment for use by the entire organization.

To ease the transition from a development environment to a production environment, it is best to use virtual host names from the start, as described in the "[Scalability Considerations](#)" section. With this strategy, you can grow your deployment without having to greatly modify the integration of your services and components.

Does your organization need high availability?

Keep in mind that implementing a high availability deployment generally doubles the need for hardware. As such, you should consider implementing cold failover strategies and becoming familiar with Oracle 10g Real Application Clusters (RAC).

Storage strategies in a high availability deployment are much different than in a standard deployment. If shared storage is needed, consider external Small Computer System Interface (SCSI) storage or a storage area network (SAN), a high-speed specialized network that interconnects different kinds of data storage devices with associated data servers or network attached storage (NAS).

The increased complexity of a high availability environment means that installation and configuration involves significantly more planning and effort than does a non-high availability environment. Management of a high availability environment is also more complex and should be allocated the appropriate amount of time.

What is your organization's existing backup solution?

Recommended backup strategies for most organizations include standing policies for regular, scheduled data backups, file system backups, and tape backups. Also, your organization might have mandated recovery times and granularity for data recovery that you'll need to consider.

Oracle Beehive Deployment Configurations

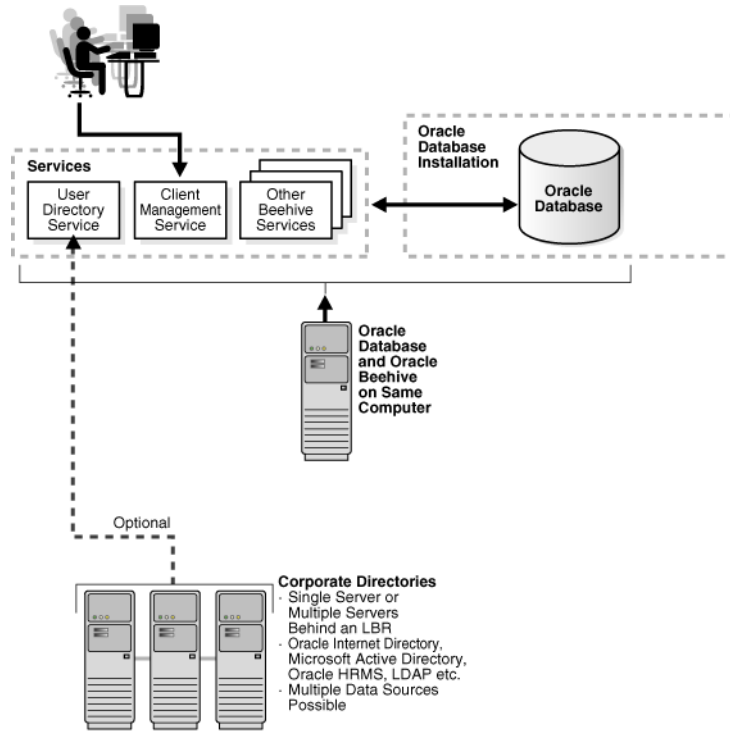
Oracle Beehive provides a flexible system deployment model that supports a variety of configurations. This section provides high-level details on the following main deployment configurations:

- [Deploying Oracle Beehive on a Single Computer](#)
- [Deploying Oracle Beehive on Multiple Computers](#)
- [Deploying Oracle Beehive Across Network Zones](#)

Deploying Oracle Beehive on a Single Computer

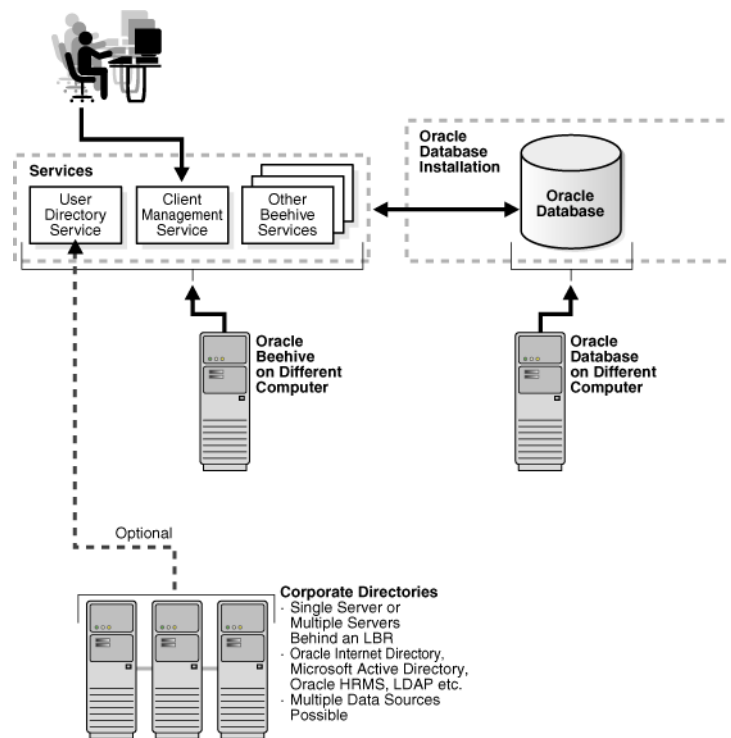
Oracle Beehive can be deployed on a single computer. In this case, the system can leverage a single Oracle Database instance located on the same computer, as illustrated by [Figure 7-1, "Oracle Beehive and Oracle Database Deployed on a Single Computer"](#), or it may leverage an Oracle Database instance that resides on another computer, as illustrated by [Figure 7-2, "Oracle Beehive and Oracle Database Deployed on Different Computers"](#). However, the latter case is more likely and, in fact, strongly recommended. The system can also leverage a database cluster, residing among several other computers, using Oracle Real Application Clusters (Oracle RAC).

Figure 7-1 Oracle Beehive and Oracle Database Deployed on a Single Computer



Additionally, Oracle Beehive provides all required user directory components so it can function as an independent system. It can also integrate with one or more external LDAP-based corporate directories such as Oracle Internet Directory or Microsoft Active Directory Server.

Figure 7-2 Oracle Beehive and Oracle Database Deployed on Different Computers



Deploying Oracle Beehive on a single computer is suitable for test environments, development environments, and small production environments. Due to the lack of its failover capabilities, this deployment scenario is unsuitable in cases where high levels of guaranteed services are required.

Deploying Oracle Beehive on Multiple Computers

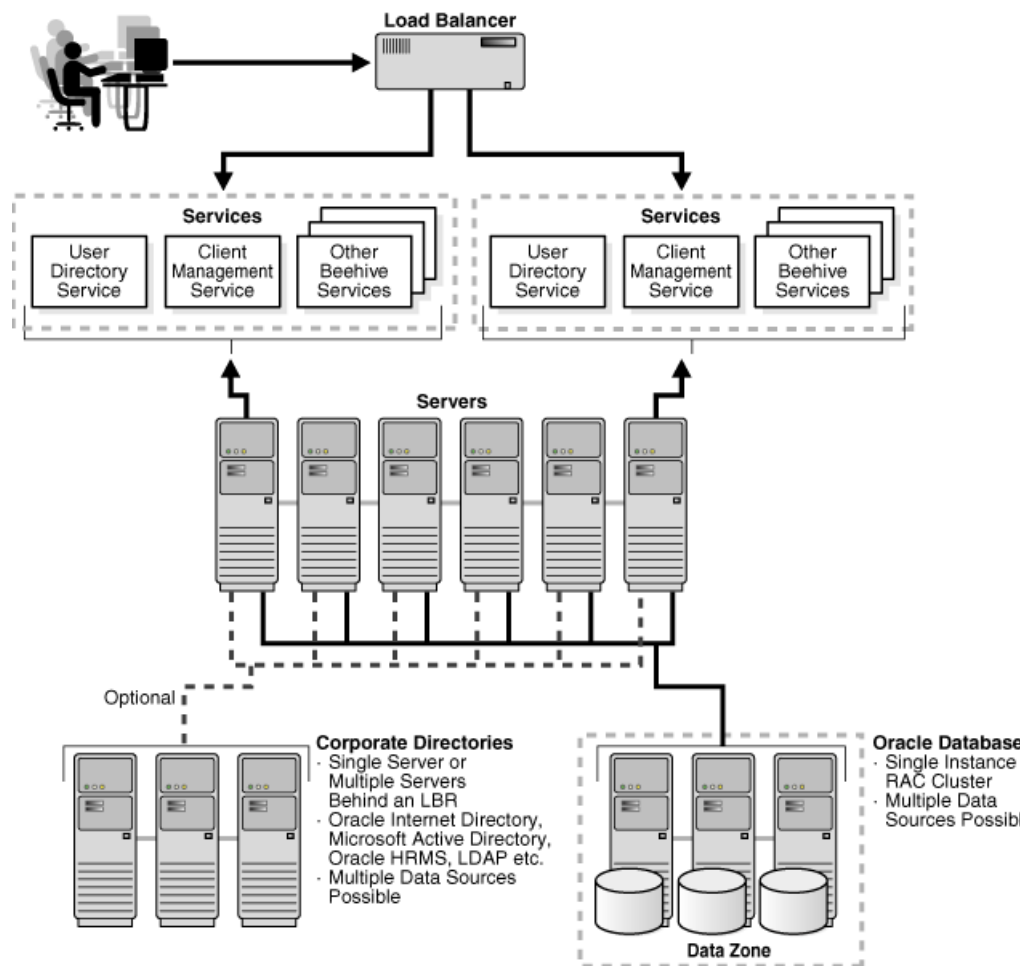
Oracle Beehive supports deployments where multiple Oracle Beehive server instances are distributed across multiple computers. In this scenario, each computer hosts one Oracle Beehive instance, which can be accessed through load balancing routers (LBRs).

Each instance runs all available services provided with Oracle Beehive. For example, if you have five computers, each with an Oracle Beehive server instance, you will have five instances each of the User Directory Service, the E-mail Service, the Workflow Service, and so forth. However, each service is instantiated as a single deployment of the service, or as a single collection of all of the available instances.

This deployment scenario requires either a single database instance or an Oracle RAC cluster. From the user directory standpoint, you can integrate Oracle Beehive with a supported external corporate directory, or you can deploy it as a standalone system that leverages its own user directory capabilities and features.

Typically, a deployment with multiple Oracle Beehive instances across multiple computers is used in test environments, large production environments, or where a higher level of service (high availability) is required. Test environments in this scenario can either be replicas of their associated production environments (recommended) or they may be scaled-down versions that mimic production environment topologies but with less hardware. Cloning of Oracle Beehive instances is supported to facilitate this process.

Figure 7-3 Multiple Oracle Beehive Instances on Multiple Computers



During the Oracle Beehive installation process, you provide the names for your Enterprise, Organization, Site, and Instance. To successfully deploy multiple Oracle Beehive instances, you must provide the same names for Enterprise, Organization, and Site for all instances. You must also specify the same Oracle Database instance for all Oracle Beehive server instances.

Deploying Oracle Beehive Across Network Zones

Oracle Beehive supports deployments across networks zones, which are used to logically split the different layers of Oracle Beehive into the following areas:

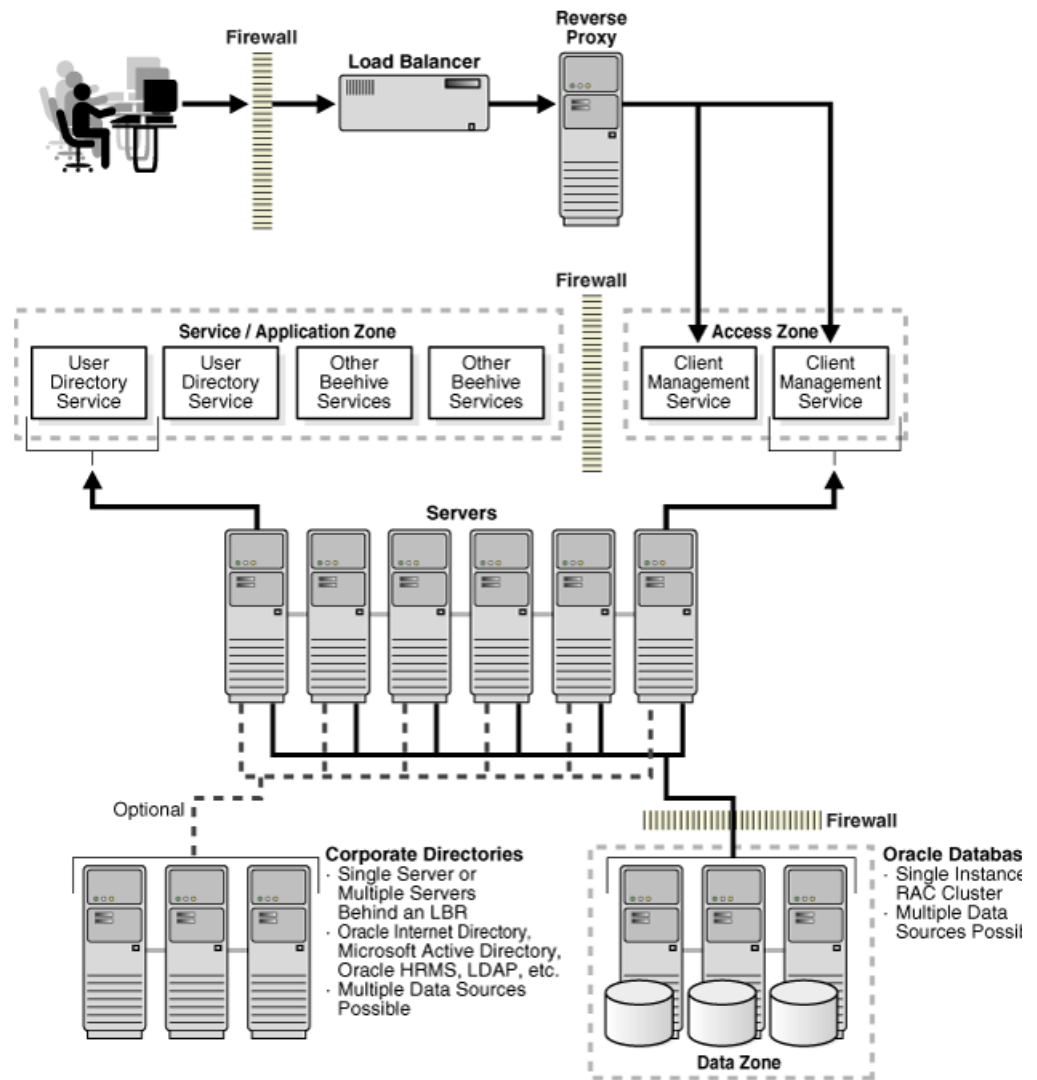
- Client Access Zone
- Application Zone
- Data Zone

Firewalls and multiple network zones are supported in this deployment scenario, providing increased security measures where user access to the system is required from corporate networks and the Internet. Network zones such as a corporate intranet or a demilitarized zone (DMZ) are also supported. Oracle Beehive services may have a dedicated system on which they run so that only the required services are exposed in the DMZ.

In this scenario, the Client Access Zone is separated from the other zones and their services, and resides in a separate network layer such as a DMZ. Firewalls may exist between the different zones, and a reverse proxy may also be present in the Client Access Zone. To provide higher levels of service (high availability), this deployment scenario may also consist of multiple computers each running an Oracle Beehive server instance. For more information, refer to "[Deploying Oracle Beehive on Multiple Computers](#)".

Typically, organizations deploy Oracle Beehive in this scenario for the increased security that it provides. This scenario protects core data (in the Data Zone) behind several layers (zones) and barriers. Similarly, this scenarios also protects application logic while seamlessly providing users needed access and functionality. Network connectivity layers are thin but, upon successful authentication, allow full access to available services.

Figure 7-4 Oracle Beehive Deployed Across Multiple Network Zones



How the Beehive Transport Infrastructure (BTI) Supports Multiple Network Zones

Deployments that leverage segregated network zones with limited connectivity between these zones is common. These requirements are an evolution of continually improving security policies and best practices. The BTI provides the flexibility to integrate well with this configuration scenario and almost any other topology requirements.

The BTI enables Oracle Beehive to accommodate any number of network buffer zones while still providing client access. This maintains open access to the system but limits the possibility of system compromise.

The BTI provides a plug-in (`mod_bti`) to Oracle HTTP Server, which allows enterprises to expose only one or two standard open ports to the Internet without limiting client access. The `mod_bti` does so by recognizing Oracle Beehive MX protocol traffic and transparently redirecting those requests to the appropriate Oracle Beehive services.

The BTI allows enterprises to place very thin network services in the DMZ and minimize exposure of the Application and Data Tiers, while providing secure client access to Oracle Beehive services.

Deploying Oracle Beehive with Supported Clients and Devices

The Oracle Beehive platform provides a unified client implementation model that supports deployments with a variety of end-user clients and devices. This chapter discusses the considerations related to deploying supported clients and devices, and includes the following topics:

- [Oracle Beehive Deployments with Oracle Beehive Integration for Outlook \(OBIO\)](#)
- [Oracle Beehive Deployments with Oracle Beehive Integration for Zimbra](#)
- [Oracle Beehive Deployments with Standards-based Clients](#)
- [Oracle Beehive Deployments with Mobile Devices](#)

Oracle Beehive Deployments with Oracle Beehive Integration for Outlook (OBIO)

Oracle Beehive Integration for Outlook (OBIO) is a Messaging Application Programming Interface (MAPI) service provider for Microsoft Outlook. OBIO extends the functionality of Microsoft Outlook by providing Outlook users with unified access to Oracle Beehive-based collaborative features and data in a familiar environment. OBIO requires installation on the computers of individual users.

To access, download, and install OBIO on users' computers, Oracle Beehive provides a download agent, which is accessible from the following location:

```
$ORACLE_HOME/beehive/bootstrap/obio/setup
```

Administrators can make the download agent available to users from internal websites. Upon execution, the download agent connects users to Oracle Beehive, challenges them for their credentials, and enables them to download and install OBIO. Users can delete the download agent once they complete the OBIO installation.

By default, Oracle Beehive provides OBIO as a zip file (.zip) in the Device Management Application Repository. The contents of the zip file include a Windows executable installation program (.msi) and an Extensible Markup Language (XML) file that describes the installation program. The structure of the file follows the XML-based metadata schema mandated by the Device Management Service.

Oracle Beehive also provides a language pack application module for OBIO, which can be accessed by administrators in the same location as the OBIO download agent. Administrators can modify the application module by adding or removing one or more languages. The module can be deployed to users through the Device Management Application Repository or another preferred method, such as by download from an intranet.

Administrators can upgrade existing OBIO versions simply by deploying newer packages to the Device Management Application Repository using `beectl`. Administrators can also remove existing packages and replace them with newer ones, as needed. After installation on user computers, OBIO automatically detects newer versions that are made available through the Device Management Service. Users can receive updates automatically upon Outlook startup or manually by clicking the update option in the About OBIO dialog window.

OBIO communicates with Oracle Beehive through the Beehive Transport Infrastructure (BTI) and the proprietary MX protocol that the BTI provides. Thus, OBIO users can either connect directly to an Oracle Beehive deployment or they can be tunneled through standard HTTPS ports.

Oracle Beehive Deployments with Oracle Beehive Integration for Zimbra

Oracle Beehive Integration for Zimbra (OBIZ) is a Web-based client that enables Oracle Beehive users to access, view, and manage their e-mail, calendars, and address books.

Oracle Beehive Integration for Zimbra supports the following Web browsers in the Windows, Linux, and Mac OS X operating systems:

- Mozilla Firefox 2.0
- Mozilla Firefox 3.0
- Microsoft Internet Explorer 6.0
- Microsoft Internet Explorer 7.0

Deploying Oracle Beehive Integration for Zimbra requires installation on an Oracle Beehive server instance, in an Oracle home designated for Zimbra only. The Oracle Beehive Integration for Zimbra installation is bundled with the Oracle Beehive installation and can be performed as part of the latter's installation procedure. Once installed and configured on an Oracle Beehive server instance, users can access and leverage Oracle Beehive Integration for Zimbra simply by launching a supported Web browser and entering the URL configured for the client.

Oracle Beehive Deployments with Standards-based Clients

Oracle Beehive supports several standards-based protocols, enabling organizations to deploy commonly-available and custom clients with the platform. Standards-based clients can provide a wide array of Oracle Beehive functionality, such as e-mail, time management, and instant messaging.

Deploying standards-based clients typically requires the following steps:

- Deciding on which standards-based clients your organization wants to deploy. For the list of clients supported by Oracle Beehive, please refer to "Standards-based Clients Supported by Oracle Beehive" in *Oracle Beehive Concepts*.
- Making available for download the installation packages of the standards-based clients that your organization wants to deploy. Typically, this can be accomplished by posting the installation packages on your organization's intranet or from another internally accessible location.
- Opening firewall ports for any standards-based services required by the clients that you plan to deploy, such as SMTP (for e-mail clients) or XMPP (for instant messaging clients). This step should be completed if you expect to support remote users, although it might not be required in cases where remote users access Oracle Beehive through virtual private networks (VPNs).

- Installing the selected clients on users' computers and devices. For details on how to install a particular standards-based client, please refer to the documentation provided with the client that you want to install.
- Configuring the installed client applications based on the settings of your Oracle Beehive deployment. For details on how to configure a supported standards-based client for Oracle Beehive, please refer to "Using Standards-based Clients" in *Oracle Beehive End-User Information*.

Note:

- For additional considerations related to deploying standards-based clients on mobile devices, please refer to "[Oracle Beehive Deployments with Mobile Devices](#)".
-
-

Oracle Beehive Deployments with Mobile Devices

Oracle Beehive enables users of supported mobile devices to send and receive collaborative data, such as e-mail, calendar synchronization, and instant messages. Deploying mobile devices with Oracle Beehive typically requires completion of the following tasks:

- Deciding on which mobile devices your organization wants to deploy. For the list of devices and device-specific features that Oracle Beehive supports, please refer to "Mobile Devices Supported by Oracle Beehive" in *Oracle Beehive Concepts*.
- Deciding on which client applications, if any, your organization wants to deploy on users' mobile devices. For the list of clients that Oracle Beehive supports, please refer to "Oracle Beehive End-User Clients" in *Oracle Beehive Concepts*.
- Uploading the selected client applications to the Device Management Application Repository. You perform this step using the `beectl` utility. For more information, please refer to the *Oracle Beehive Administrator's Guide*.
- Opening firewall ports for any protocol services required by the devices and clients that you plan to deploy, such as Push Internet Message Access Protocol (P-IMAP) and Open Mobile Alliance Data Synchronization (OMA-DS).
- Registering the selected mobile devices with Oracle Beehive and configuring them based on the settings of your deployment. For details on how to register and configure supported devices for Oracle Beehive, please refer to "Configuring Mobile Devices" in *Oracle Beehive End-User Information*.
- Installing the selected clients on users' devices. For details on how to install a particular client, please refer to the documentation provided with the client that you want to install.
- Configuring the installed client applications based on the settings of your Oracle Beehive deployment. For considerations on deploying standards-based mobile clients, please refer to "Configuring Mobile Clients" in *Oracle Beehive End-User Information*. For considerations on deploying Microsoft Windows Mobile-based clients, please refer to "[Deploying Microsoft Windows Mobile-based Clients Supported by Oracle Beehive](#)".
- Enabling the Oracle Beehive SMS Delivery Channel, if your organization wants to leverage the network of a Short Message Service (SMS) service provider. For more information, please refer to "[Enabling the Oracle Beehive Short Message Service \(SMS\) Delivery Channel](#)".

Deploying Microsoft Windows Mobile-based Clients Supported by Oracle Beehive

For Microsoft Windows Mobile devices, Oracle Beehive provides a plug-in for Microsoft Windows Mobile Outlook that enables real-time push mail. To install, provision, and manage the plug-in, Oracle Beehive provides the Mobile Management Console. Users must first install the Mobile Management Console, provided as a cabinet file (.cab), on their mobile devices through one of the following methods:

- **Over-the-air (OTA):** Users open a pre-assigned Web page or URL from their mobile devices, or they receive a Short Message Service (SMS) message that contains the URL, enabling them to access, download, and install the .cab file. For more information on these options, please refer to "Over-the-Air Configuration Using SMS" and "Over-the-Air Configuration Without SMS" in *Oracle Beehive End-User Information*.
- **Active synchronization (ActiveSync):** Users download the .cab file to their computers through a Windows-based utility provided with Oracle Beehive. Users then synchronize their mobile devices with their computers to install the .cab file. For more information on this option, please refer to "Configuration Using ActiveSync" in *Oracle Beehive End-User Information*.
- **Storage card:** Administrators or support personnel download the .cab file to their computers through a Windows-based utility provided with Oracle Beehive. Then, they run the utility from the command line to install the .cab file on storage cards. Users can then insert the storage cards into their devices.

After the Mobile Device Management Console is installed, it automatically starts up, connects to the Oracle Beehive Mobile Device Management Service, and requests a list of all available mobile software for the user. This list contains the default Mobile Outlook plug-in but it can also contain other third-party mobile software managed by Oracle Beehive. The user may then choose which software to install on the mobile device, including the Mobile Outlook plug-in.

To upgrade the Mobile Outlook plug-in or other third-party mobile software managed by Oracle Beehive, users can periodically check for updates on their mobile devices or they can wait for their Oracle Beehive administrators to provide alerts informing them of available updates.

Enabling the Oracle Beehive Short Message Service (SMS) Delivery Channel

The Oracle Beehive SMS Delivery Channel is the component of the Message Delivery Service that enables delivery of SMS messages. The SMS Delivery Channel supports the Short Message Peer-to-Peer (SMPP) protocol, which is a telecommunications industry standard supported by most SMS service providers.

By default, the Oracle Beehive SMS Delivery Channel is configured to use the Verisign Intelligent Messaging Network. However, organizations may choose to leverage the network of any SMS service provider that supports SMPP. Organizations may also leverage the Oracle iAS Wireless XMS interface, which was the default interface for Oracle Collaboration Suite 10g and is also supported by Oracle Beehive.

The steps required to configure and enable the SMS Delivery Channel depend on which SMS service provider you choose. For example, if you choose the Verisign Intelligent Messaging Network, you must first subscribe to it (to request a free trial, please contact Verisign). Once you activate your Verisign subscription, you will need to configure and enable the Oracle Beehive SMS Delivery Channel for it. This includes configuring your SMPP settings, setting your SMS mode to "SMPP", and enabling the SMS Delivery Channel. To perform these tasks you use the `beect1` utility.

For more information on the specific steps required to configure the SMS Delivery Channel for your chosen SMS provider, please contact Oracle Support.

Index

A

access, 5-1, 5-5
administration, 1-5
 tools
 Oracle Beehive Administration Console, 3-2
Ancillary Tier, 2-1, 2-3
Application Tier, 2-1, 2-2, 2-4
Application Zone, 7-4
availability, 6-2

B

backup, 6-4
beectl, 8-2
Beehive Transport Infrastructure (BTI), 2-4, 5-2, 5-3,
 7-6, 8-2
Berkeley Internet Name Domain (BIND), 5-3
BIND
 See Berkeley Internet Name Domain (BIND)
BPEL
 See Business Process Execution Language (BPEL)
BTI
 See Beehive Transport Infrastructure
 See Beehive Transport Infrastructure (BTI)
business events, 2-5
Business Process Execution Language (BPEL), 2-3

C

CalDAV
 See Calendaring Extensions to WebDAV (CalDAV)
Calendaring Extensions to WebDAV (CalDAV), 2-2
client
 standard protocol, 8-2
Client Access Zone, 7-4
Client Tier, 2-1, 2-2
clients
 mobile devices, 8-3
 Oracle Beehive Integration for Outlook
 (OBIO), 8-1
cloning, 7-3
Code Schema
 about, 2-6
configuration, 1-4
content switches, 5-4

customization, 1-4

D

Data Schema
 about, 2-6
Data Tier, 2-1, 2-3, 2-4
Data Zone, 7-4
Database Access Framework, 2-3, 2-4
 about, 2-4
demilitarized zones (DMZs), 1-6, 2-5, 5-2, 5-3, 5-5,
 5-6, 6-1, 6-3, 7-6
deploying
 across network zones, 7-4
 in stages, 6-3
 on a single computer, 7-1
 on multiple computers, 7-3
deployment
 prerequisites, 1-2
 stages, 1-2
Device Management Repository, 8-1, 8-2
DMZ
 See demilitarized zones (DMZs)
DNS
 See Domain Name System (DNS)
Domain Name System (DNS), 5-2, 5-4, 5-5
 servers, 5-3, 5-4

E

enterprises, 1-7
Event Framework, 2-4
 about, 2-5
 extending, 2-5
events
 business events, 2-5
Extensible Messaging and Presence Protocol
 (XMPP), 2-2

F

File Transfer Protocol (FTP), 2-2
firewalls, 5-2, 5-4, 5-5

H

high availability, 6-1, 6-4, 7-5

HTTP

See HyperText Transfer Protocol (HTTP)

HTTPS

See Hypertext Transfer Protocol Over Secure Socket Layer

See Secure HyperText Transfer Protocol (HTTPS)

HyperText Transfer Protocol (HTTP), 5-2, 5-4

Hypertext Transfer Protocol Over Secure Socket Layer (HTTPS), 3-2

I

IBM Tivoli, 2-4, 4-3

installation, 1-4, 7-4

instances, 1-7

Internet Message Access Protocol (IMAP), 2-2, 5-5

J

J2EE

See Java 2 Platform Enterprise Edition (J2EE)

Java 2 Platform Enterprise Edition (J2EE), 2-1

Java Database Connectivity (JDBC), 2-4

JDBC

See Java Database Connectivity (JDBC)

L

launching the system, 1-5

load balancers, 5-4, 5-5, 7-3

M

mail relays, 5-2, 5-4, 5-5

maintenance, 1-5

Microsoft Active Directory, 2-4

Microsoft Active Directory Server, 4-3

Microsoft Exchange Server, 2-4

Microsoft Outlook, 1-1, 2-2, 2-4

Microsoft Windows, 8-4

Microsoft Windows Mobile Outlook, 8-4

plug-in for, 8-4

Mobile Device Management Console, 8-4

Mobile Device Management Service, 8-4

mobile devices

deploying, 8-3

Microsoft Windows-based, 8-4

mod_bti, 7-6

multiple computer deployments, 7-3

multiplexor protocol (MX), 2-3, 7-6, 8-2

MX

See multiplexor protocol (MX)

N

NAS

See network attached storage (NAS)

NAT

See Network Address Translation (NAT)

Network Address Translation (NAT), 5-4

network attached storage (NAS), 6-4

network capacity, 5-6, 6-2

network services, 5-6

network traffic

external, 5-2

internal, 5-2

network zone deployments, 7-4

network zones, 1-6, 7-4

Application Zone, 7-4

Client Access Zone, 7-4

Data Zone, 7-4

nodes, 1-6

O

OID

See Oracle Internet Directory (OID)

OMA-DS

See Open Mobile Alliance Data Synchronization

Open Mobile Alliance Data Synchronization

(OMA-DS), 2-2, 8-3

Oracle Application Server, 2-1, 2-2

Oracle Beehive

schemas, 2-4

about, 2-5

Code Schema, 2-6

Data Schema, 2-6

tiers

Ancillary Tier, 2-3

Application Tier, 2-2

Client Tier, 2-2

Data Tier, 2-3

Oracle Beehive Administration Console, 3-2

Oracle Beehive Integration for Outlook (OBIO), 2-4,

8-1

deploying, 8-1

download agent, 8-1

upgrading, 8-2

Oracle Beehive Integration for Zimbra (OBIZ), 8-2

Oracle BPEL Process Manager, 2-3

Oracle Database, 2-1, 2-3, 6-2, 7-1, 7-4

Oracle HTTP Server, 5-2, 5-3, 7-6

Oracle Internet Directory, 4-3

Oracle Internet Directory (OID), 2-3

Oracle Real Application Clusters (Oracle RAC), 2-3,

7-1

nodes

adding, 2-4

removing, 2-4

Oracle Secure Enterprise Search, 2-3

organizations, 1-7

P

P-IMAP

See Push Internet Message Access Protocol

policies

network, 5-3

security, 5-3
Push Internet Message Access Protocol
(P-IMAP), 2-2, 8-3

R

RAC
 See Oracle Real Application Clusters (Oracle RAC)
recovery, 6-2
reverse proxies, 5-2, 5-4
routers, 5-5

S

SAN
 See storage area network (SAN)
scalability, 6-2
schemas
 about, 2-5
 Oracle Beehive, 2-4
SCSI
 See Small Computer System Interface (SCSI)
Secure HyperText Transfer Protocol (HTTPS), 5-2,
 5-4, 5-5
Secure Sockets Layer (SSL), 5-2, 5-4, 5-5
security, 5-1, 6-3
 policies, 5-3
 spam and virus filters, 5-5
servers, 1-6
services, 1-6, 5-5
Simple Mail Transfer Protocol (SMTP), 2-2, 5-4, 5-5
single computer deployments, 7-1
sites, 1-7
Small Computer System Interface (SCSI), 6-4
SMTP
 See Simple Mail Transfer Protocol (SMTP)
spam, 5-5
 services, 5-5
SSL
 See Secure Sockets Layer (SSL)
standard protocol clients
 deploying, 8-2
storage
 network attached storage (NAS), 6-4
 Small Computer System Interface (SCSI), 6-4
 storage area network (SAN), 6-4
storage area network (SAN), 6-4
Sun Java Directory Server, 2-4
Sun Java System Directory Server, 4-3
Symantec, 2-4
Symantec Scan Engine, 2-4
system administration, 1-5
system configuration, 1-4

T

testing, 1-5
tiers
 Ancillary Tier, 2-3
 Application Tier, 2-2, 2-4
 Client Tier, 2-2

Data Tier, 2-3, 2-4
TLS
 See Transport Layer Security (TLS)
training, 1-5
Transport Layer Security (TLS), 5-2

V

Virtual Private Networks (VPNs), 5-5
viruses, 5-5
VPN
 See Virtual Private Networks (VPNs)

W

Web-based Distributed Authoring and Versioning
(WebDAV), 2-2
workload, 6-2

Z

zones
 network, 1-6

