Oracle® Communications Indexing and Search Service

Installation and Configuration Guide

Release 1.0.5

E67901-03

August 2017

Oracle Communications Indexing and Search Service Installation and Configuration Guide, Release 1.0.5

E67901-03

Contents

3 Indexing and Search Service System Requirements

4 Indexing and Search Service Pre-Installation Tasks

5 Installing Indexing and Search Service

6 Configuring Indexing and Search Service with Multiple Hosts

7 Configuring Indexing and Search Service Web Services Proxy

8 Configuring Indexing and Search Service for High Availability

11 Upgrading Indexing and Search Service

12 Uninstalling Indexing and Search Service

13 Installing Patches

A Indexing and Search Service Scripts

B commpkg Reference

Preface

This guide provides instructions for installing and configuring Oracle Communications Indexing and Search Service.

Audience

This document is intended for system administrators or software technicians who install and configure Indexing and Search Service. This guide assumes you are familiar with the following topics:

- Oracle Communications Messaging Server
- Oracle GlassFish Server
- Oracle GlassFish Message Queue
- System administration and networking

Related Documents

For more information, see the following documents in the Indexing and Search Service documentation set:

- *Indexing and Search Service Release Notes*: Describes the fixes, known issues, troubleshooting tips, and required third-party products and licensing.
- *Indexing and Search Service System Administrator's Guide*: Provides instructions for administering Indexing and Search Service.
- *Indexing and Search Service Security Guide*: Provides guidelines and recommendations for setting up Indexing and Search Service in a secure configuration.

Documentation Accessibility

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

Access to Oracle Support

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

1

Indexing and Search Service Installation Overview

This chapter provides an overview of the Oracle Communications Indexing and Search Service installation process.

Overview of Indexing and Search Service Installed Components

During the installation process, you install and configure the following components:

- Java
- Message Queue
- Oracle GlassFish Server
- (Optional): Apache HTTP Server version 2, for multiple host deployments (used as web container for Indexing and Search Service back-end hosts)
- Indexing and Search Service

Indexing and Search Service assumes that you have already installed Oracle Communications Messaging Server and Oracle Directory Server Enterprise Edition. If your site does not currently have Messaging Server and Directory Server deployed and you need to install these products, see *Messaging Server Installation and Configuration Guide* for instructions.

To use Indexing and Search Service with Oracle Communications Convergence, Indexing and Search Service assumes that you have already installed Convergence. If your site does not currently have Convergence deployed and you need to install it, see *Convergence Installation and Configuration Guide* for instructions.

Overview of the Indexing and Search Service Installation Procedure

The installation procedure follows these steps:

1. Plan your installation. When planning your installation, do the following:
   - Determine the scale of your implementation, for example, a small development system, or a large production system.
   - Determine how many physical machines you need, and which software components to install on each machine.
   - Plan the system topology, for example, how the system components connect to each other over the network.
2. Review system requirements. System requirements include:

  - Hardware requirements, such as disk space.
  - System software requirements, such as operating system (OS) versions and OS patch requirements.
  - Information requirements, such as IP addresses and host names.

3. Install and configure software upon which Indexing and Search Service is dependent, including:
   - Java
   - Message Queue
   - GlassFish Server
   - (Optional): Apache HTTP Server version 2, for multi-host deployments (used as web container for Indexing and Search Service back ends)
4. Install and configure Indexing and Search Service.
5. (Optional) Configure additional Indexing and Search Service front ends and back ends for a multiple host deployment.
6. Perform post-installation configuration tasks.
7. Verify the installation.

After Indexing and Search Service is installed, you might perform additional security-related tasks. For more information, see *Indexing and Search Service Security Guide*.

Indexing and Search Service Installation Options

You install Indexing and Search Service in either interactive or silent mode. When you run the installer in silent mode, you are running a non-interactive session. The installation inputs are taken from the following sources:

- A silent installation file
- Command-line arguments
- Default settings

You can use silent mode to install multiple instances of the same software component and configuration without having to manually run an interactive installation for each instance.

Ensuring a Successful Indexing and Search Service Installation

Only qualified personnel should install the product. You must be familiar with the UNIX operating system and Oracle GlassFish Server. You should be experienced with installing Java-related packages.

Follow these guidelines:

- As you install each component, for example, GlassFish Server, verify that the component installed successfully before continuing the installation process.
- Pay close attention to the system requirements. Before you begin installing the software, make sure your system has the required base software. In addition, ensure that you know all of the required configuration values, such as host names and port numbers.

- As you create new configuration values, write them down. In some cases, you might need to reenter configuration values later in the procedure.

Directory Placeholders Used in This Guide

Table 1–1 lists the placeholders that are used in this guide:

Table 1–1 Indexing and Search Service Directory Placeholders

Placeholder	Directory
IndexSearch_home	Specifies the installation location for the Indexing and Search Service software. The default is **/opt/sun/comms/jiss**.
MessagingServer_home	Specifies the installation location for the Messaging Server software. The default is **/opt/sun/comms/messaging64**.
GlassFish_home	Specifies the installation location for the Oracle GlassFish Server software. The default is **/opt/glassfish3/glassfish**.
Convergence_home	Specifies the installation location for the Oracle Communications Convergence software. The default is **/opt/sun/comms/iwc**.

2

Planning Your Indexing and Search Service Installation

This chapter provides information about planning your Oracle Communications Indexing and Search Service installation. It also describes the Indexing and Search Service logical and physical architectures.

About Indexing and Search Service

Indexing and Search Service is a general purpose indexing and search service for Oracle Communications Messaging Server, enabling real-time indexing and fast search for email. All IMAP email clients can benefit from Indexing and Search Service's fast body search and attachment content search. In addition, Indexing and Search Service is integrated with the Oracle Communications Convergence Web client, enabling unique attachment browser capability.

The Convergence Web client provides a system folder, **Attachments**, under the mail tree. When users click the **Attachments** folder, they are able to view all the attachments in their inbox. This view enables users to browse attachments and search for an attachment by using filters. Indexing and Search Service provides a plug-in to Convergence that generates searches directly to Indexing and Search Service for attachment search.

Why Use Indexing and Search Service

An indexing and search service is necessary in today's environment to quickly find information from among large and ever growing data sets. Indeed, as data quotas approach infinity, email stores have become user archives for presentations, videos, audio files, and much more. Search is now the accepted method to navigate these stores.

Indexing and Search Service provides the following capabilities that meet an organization's search needs:

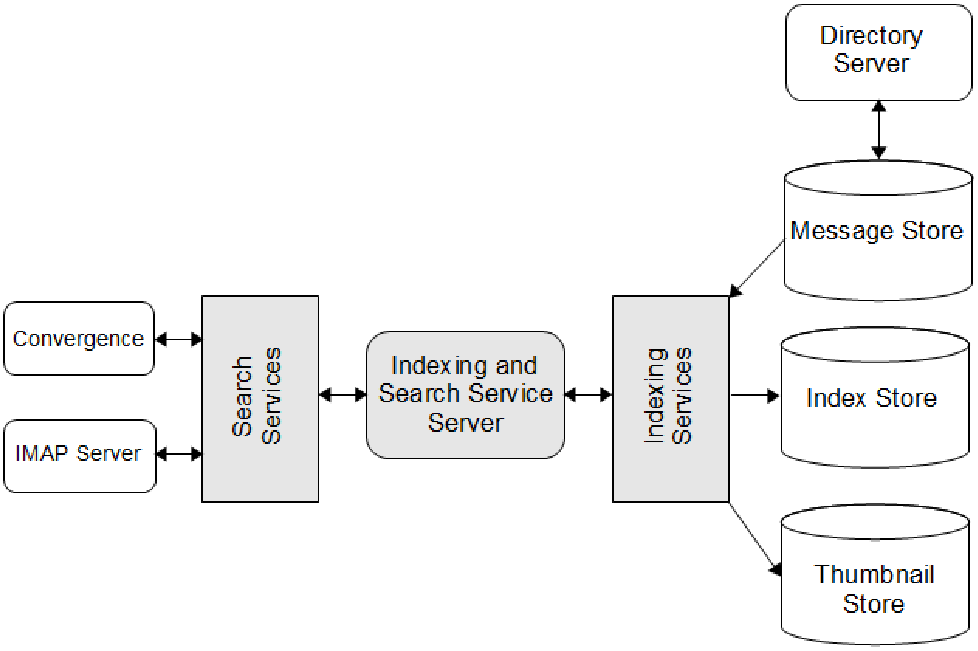
- Uses standards-based web services search interface
- Updates indexes in real-time as content changes
- Analyzes attachment content and makes it searchable
- Generates and returns attachment thumbnails for visual representation of search results
- Offloads Messaging Server from conducting performance-intensive body search (Indexing and Search Service search is orders of magnitude faster)

- Enables multi-folder search, which is inherently fast

Indexing and Search Service Logical Architecture

Figure 2–1 shows a high-level overview of the Indexing and Search Service software architecture.

Figure 2–1 Indexing and Search Service Architecture

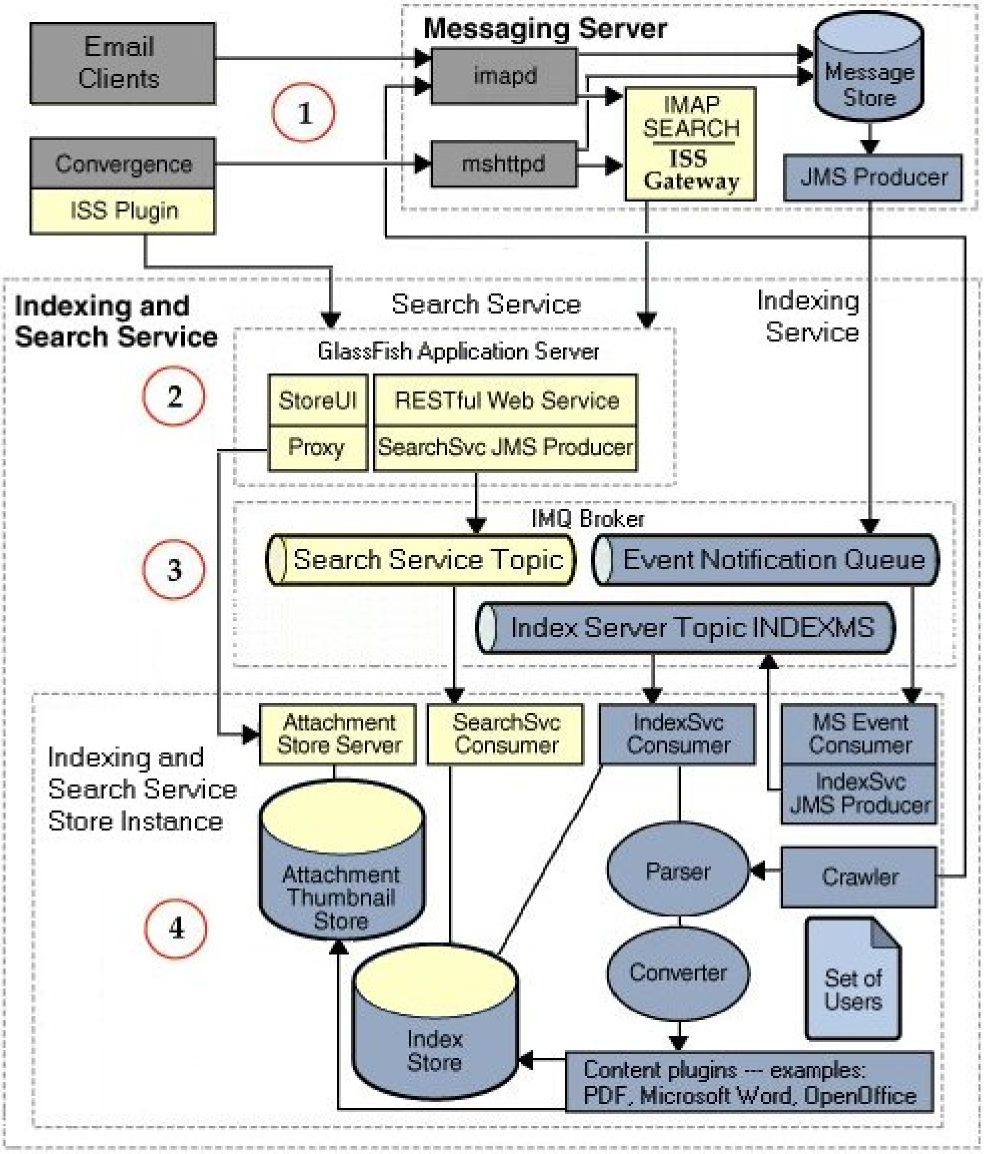


This figure shows that Indexing and Search Service is composed of an *indexing service* and a *search service*. The indexing service indexes email data in real time. Indexing is provided through web services, enabling you to index arbitrary data. Indexing and Search Service clients consume RESTful web services that provide the search capabilities. Components communicate over Java Messaging System (JMS) enabling distribution of components across multiple machines.

The way in which Indexing and Search Service handles attachments helps to improve search performance. Attachment files to emails are not saved to the index store. Instead, Indexing and Search Service provides search results that include body part information for pointing to the attachment file residing on the Messaging Server message store. Additionally, the Indexing and Search Service attachment store contains only thumbnail images of attachments.

Figure 2–2 shows a more detailed look at components of the Indexing and Search Service software architecture.

Figure 2–2 Indexing and Search Service Detailed Architecture



This figure labels the major layers of the Indexing and Search Service architecture. In the first layer, Messaging Server manages the message store, and email clients, such as Convergence and Thunderbird, manage and search this content. In the second layer, GlassFish Server hosts the Search Service programs, along with the RESTful web API. In the third layer, a JMQ broker hosts the search queue and email update notifications for the indexing queue. In the fourth layer, the index store manages the indexing of the message store content, and the store instance creates and manages the attachment thumbnail images of the message store attachment content.

Additionally, a JMQ broker is used for Messaging Server notifications and for internal Indexing and Search Service communication. In a simple configuration, you use one JMQ broker. For more complex deployments, you can use a separate JMQ broker running on another host or on the Messaging Server host for the Messaging Server notifications.

How Indexing and Search Service Performs Searches

From an architectural standpoint, IMAP clients perform searches on the Indexing and Search Service store by first connecting to the Messaging Server IMAP daemon. The IMAP SEARCH Indexing and Search Service gateway component of Messaging Server determines if the IMAP search should be handled by Indexing and Search Service. For information, see the topic on how Indexing and Search Service and IMAP SEARCH handle searches in *Indexing and Search Service System Administrator's Guide*.

Planning Considerations

This section contains the following planning topics you must consider before installing Indexing and Search Service.

Messaging Server Considerations

Indexing and Search Service requires Messaging Server to be installed. See "Required Software" for supported software versions.

Message Queue Considerations

Indexing and Search Service requires Message Queue to be installed for notification services. See "Required Software" for supported software versions.

GlassFish Server Considerations

Indexing and Search Service search servlet requires GlassFish Server as a web container in which to run. See "Required Software" for supported software versions.

Apache HTTP Server Considerations

In a multiple host architecture, where the web tier is running on separate systems from the indexing tier, Apache HTTP Server 2 must be installed on the systems in the indexing tier. See "Required Software" for supported software versions.

Web Services Proxy Considerations

When deploying Indexing and Search Service with Convergence in a multiple host deployment, you must install and configure the Indexing and Search Service Web Services Proxy (**isshttpd**). The Indexing and Search Service Web Services Proxy enables Convergence to proxy requests from client search queries to the correct Indexing and Search Service web services host. For more information, see "Configuring Indexing and Search Service Web Services Proxy".

Deciding on Which Indexing and Search Service Components to Install

Indexing and Search Service software consists of the following components:

- Web
- Message Queue
- LDAP
- Messaging Server
- Index

- Watcher

Depending on your Indexing and Search Service architecture, you might need to configure all Indexing and Search Service components on a single host, or spread components across multiple hosts. Use this section to understand which components you need on which hosts. You run the Indexing and Search Service **setup** script to configure Indexing and Search Service components.

Web Component

The Indexing and Search Service Web component (**web**) contains the front-end web server access to the search services. You configure the Web component on your GlassFish Server hosts.

The **setup** script installs and configures the following three JAR files on GlassFish Server:

- **rest**: This component is the RESTful API. You access the RESTful service at the following URL:

  ```
  http://GlassFish_host:port/rest
  ```

- **searchui**: This component is a sample search UI. You access the sample search UI at the following URL:

  ```
  http://GlassFish_host:port/searchui
  ```

- **storeui**: This component controls access to the attachment store. You access it at the following URL:

  ```
  http://GlassFish_host:port/store
  ```

Message Queue Component

The Indexing and Search Service Message Queue component (**jmq**) provides notification services. You can configure the Message Queue component on any host that is accessible from both the Messaging Server host and the Indexing and Search Service host. Most deployments choose to configure notification services on the host running Indexing and Search Service.

Directory Server Component

The Indexing and Search Service LDAP component (**ldap**) provides Directory Server for Java Naming and Directory Interface (JNDI) lookups. You configure the LDAP component on the Indexing and Search Service host, not on the Directory Server host (which typically serves your entire deployment). The Indexing and Search Service LDAP component enables the external Directory Server to host Indexing and Search Service's JDNI lookups.

Indexing Component

The Indexing and Search Service component (**index**) provides the following core index and search services that are required for normal operation:

- Index Service (**indexSvc**) – Bootstraps new users and indexes incoming content.
- Search Service (**searchSvc**) – Searches index and returns results.
- JMQ Consumer (**jmqconsumer**) – Listens to the JMQ on the message store and notifies **indexSvc** if new content requires indexing.

- Util Service (**utilSvc**) – Provides utility services to Index Service.

You configure the Indexing component on your Indexing and Search Service back-end hosts.

Firewall Requirements

If you deploy a firewall between your Messaging Server hosts and Indexing and Search Service hosts, you must open an additional port on the firewall for Message Queue. That is, Message Queue requires two ports, port 7676 and a second port that you manually define. See the topic on connecting through a firewall in *Oracle GlassFish Server Administration Guide* at:

`https://docs.oracle.com/cd/E19798-01/821-1794/gcuhq/index.html`

Disk Space Requirements

Indexing and Search Service takes additional disk space to store its indexes. For more information, see "Hardware Requirements".

Using Oracle ZFS

ZFS provides the following features that make it ideal for Indexing and Search Service:

- Snapshot backup
- Enables the use of less expensive SATA drives
- Built-in volume manager that enables you to grow file systems dynamically

For information on file system and ZFS configuration recommendations, see *Indexing and Search Service System Administrator's Guide*.

Multiple Host Deployment Architecture

You can deploy Indexing and Search Service components on a single host or in a multiple host architecture. If you use a multiple host deployment, the logical architecture consists of the following tiers and components:

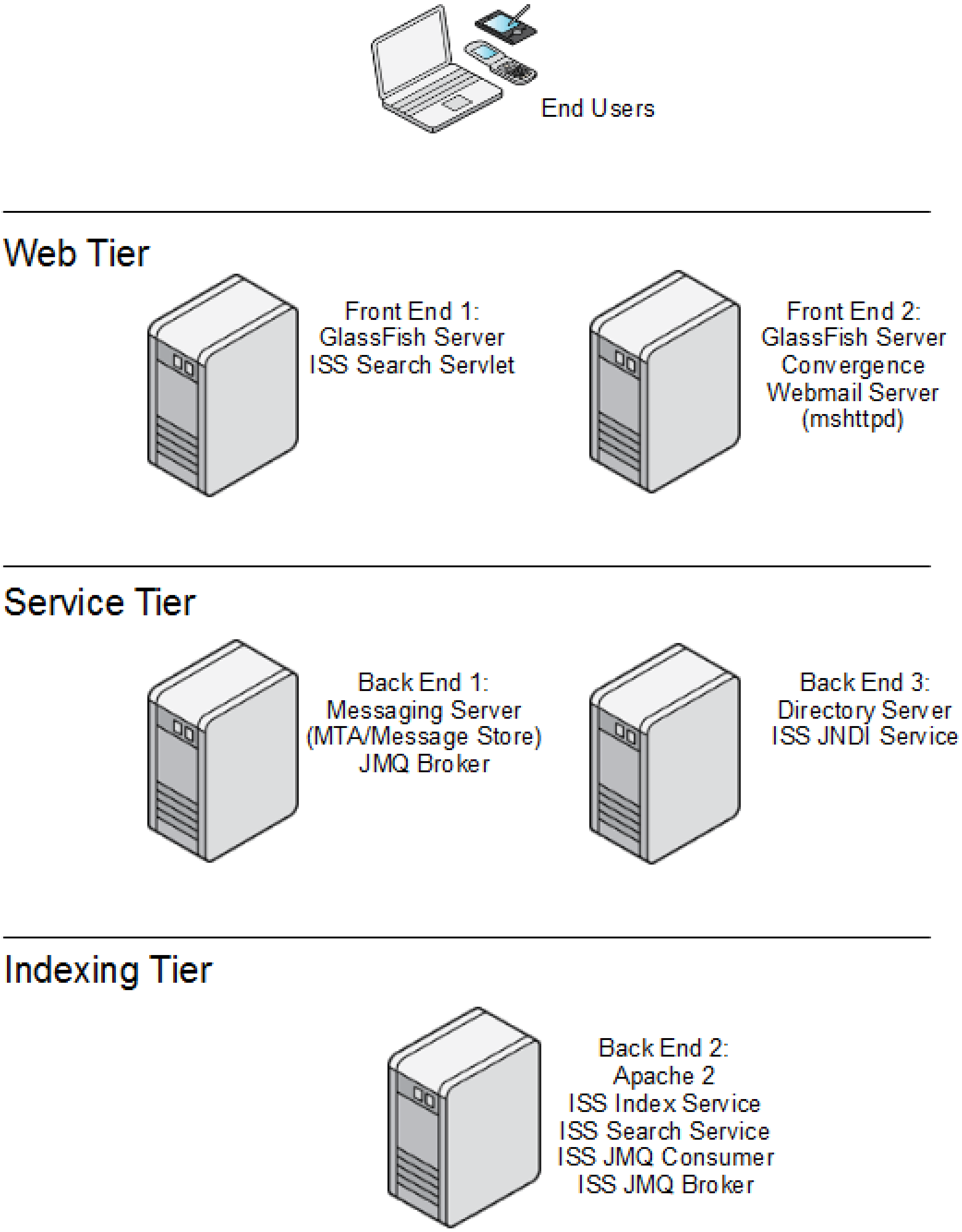
- Web Tier
  - GlassFish Server
  - Indexing and Search Service components: **storeui** (enables access to images and attachments); **searchui** (JMAKI interface for demo purposes); **rest** (RESTful API)
- Service Tier
  - Directory Server, for JNDI lookups on multiple host and cluster deployments; most likely you would use whatever Directory Server you are currently using in your deployment. (Single server or standalone deployments use file-based JNDI lookups.) Indexing and Search Service must use the same Directory Server that Messaging Server uses, whether you use Directory Server based or file-based JNDI.
  - Message Queue
- Indexing Tier

- Apache HTTP Server (where the web tier is running on separate systems from the indexing tier)
- Indexing and Search Service components: **searchSvc** (for searching); **indexSvc** (for adding new users and processing mail server events); **jmqconsumer** (listens to the Messaging Server Message Queue for events to process and sends them to **indexSvc**)

> **Note:** If you deploy Indexing and Search Service in a multiple host tiered architecture, you must run the installer multiple times to install Indexing and Search Service software on each zone or host. Likewise, you also must run the Indexing and Search Service **setup** initial configuration script multiple times on each zone or host.

Figure 2–3 shows how to separate the Indexing and Search Service components across a multiple host deployment.

Figure 2–3 Indexing and Search Service Multiple Host Architecture



In the preceding figure, Indexing and Search Service is deployed onto multiple hosts. The Indexing and Search Service web search is deployed on host **Front End 1** and the indexing service is deployed on host **Back End 2**. Hosts **Front End 2**, **Back End 1**, and **Back End 3**, complete the deployment with Convergence, the Messaging Server store, and the Directory Server, respectively.

When you deploy Indexing and Search Service in a multiple host environment, where the web tier is running on separate systems from the indexing tier, the back end requires Apache HTTP Server.

High Availability

Indexing and Search Service provides the ability to make its search component highly available with the Cluster Search Service and access the index and attachment thumbnail stores through NFS. When Indexing and Search Service search is unavailable from an Indexing and Search Service web node, the clients' search requests are redirected to another Indexing and Search Service web node that accesses the HA NFS and locates the appropriate index. Thus, the Indexing and Search Service search component can fail without an effective loss of the overall search functionality. For more information, see "Configuring Indexing and Search Service for High Availability".

Using Load Balancing

In a large deployment, you can load-balance the Indexing and Search Service URL by using either a hardware load balancer or a DNS type of load balancing. The load balancer distributes requests to the GlassFish Server instances running Indexing and Search Service.

3

Indexing and Search Service System Requirements

This chapter describes the hardware, operating system, and software requirements for installing Oracle Communications Indexing and Search Service.

Software Requirements

Indexing and Search Service search servlet is installed on an Oracle GlassFish Server domain.

Supported Operating Systems

Table 3–1 lists operating systems that support Indexing and Search Service.

Table 3–1 Supported Operating Systems

Operating System	CPU	Required Patches
Oracle Solaris 10 and 11	SPARC, x64	See the Oracle Solaris documentation for patch information.
Oracle Linux 6 and Red Hat Enterprise Linux 6 64-bit	x64	See the Oracle Linux and Red Hat Enterprise Linux documentation for patch information.

Minimal Solaris 10 Installation Requirements

The following Oracle Solaris 10 packages are required for Indexing and Search Service:

- **SUNWgcmn**
- **SUNWgtar**
- **SUNWxcu4**
- **SUNWj5rt**: If remote installs are needed, **SUNWwgetr** and **SUNWwgetu** are also required. Confirm that they are installed by using the **pkginfo** command. Multiple host deployments require that **SUNWapch2d**, **SUNWapch2r**, and **SUNWapch2u** be installed on Indexing and Search Service indexing nodes. If they are not installed, install them from the Solaris media before installing Indexing and Search Service.

The following Solaris JumpStart profile represents the minimal set of packages required for Indexing and Search Service.

```
##
## Minimum Required Software for Reduced Networking Support
##
cluster SUNWCrnet

##
## Additional basic system services
##
# SSH Daemon, Tools and Dependencies
package SUNWgss add
package SUNWgssc add
package SUNWsshcu add
package SUNWsshdr add
package SUNWsshdu add
package SUNWsshr add
package SUNWsshu add
# NTP
package SUNWntpr add
package SUNWntpu add
# System Admin Tools (showrev)
package SUNWadmfw add
package SUNWadmfr add

##
## Indexing and Search Service Requirements
##
# GNU tar and Dependencies
package SUNWgcmn add
package SUNWgtar add
# XCU4 Utilities - ISS requires for awk and id
package SUNWxcu4 add
# Java 5 Run Time
package SUNWj5rt add
# wget - required only for remote ISS installs
package SUNWwgetr add
package SUNWwgetu add
```

Minimal Linux Installation Requirements

Linux installations can run on top of the install group **@Base** in your **kickstart.cfg** file. The following RPMs must be installed on Red Hat Enterprise Linux 5:

- **rpm-build**
- **elfutils**
- **elfutils-libs**
- **compat-libstdc++-33**
- **ntp**
- **sharutils**
- **compat-libstdc++-296**
- **libxml2**
- **xinetd**

Multiple host deployments require that the **httpd** RPM be installed on Indexing and Search Service indexing nodes.

Required Software

Table 3–2 lists the software required for installing and running Indexing and Search Service.

Table 3–2 Software Requirements

Product	Version	Notes
Oracle Directory Server Enterprise Edition	6.x, 7, 11gR1 Patch Set 2 (11.1.1.7.0)	If doing a fresh installation, use 11gR1. Should have been installed as part of Messaging Server installation.
Oracle GlassFish Server	3.1.2.8	Required as the web container. Download the patch from My Oracle Support at: `https://support.oracle.com`
Message Queue	4.4 Update 1	The Indexing and Search Service Installer can install a standalone Message Queue as a shared component for Messaging Server and Indexing and Search Service. Or, you can use an embedded Message Queue that comes with GlassFish Server software.
Oracle Communications Messaging Server	7 Update 3 or greater	Must be installed before installing Indexing and Search Service.
Java	7	Be sure to install the latest security update.

Hardware Requirements

The number and configuration of the systems that you employ for your Indexing and Search Service installation depends on the scale and the kind of deployment you have planned.

Note: The sizing estimates in this section assume proper application configuration and tuning, in a manner consistent with leading practices of Oracle Communications consulting and performance engineering. This information is provided for informational purposes only and is not intended to be, nor shall it be construed as a commitment to deliver Oracle programs or services. This document shall not form the basis for any type of binding representation by Oracle and shall not be construed as containing express or implied warranties of any kind. You understand that information contained in this document will not be a part of any agreement for Oracle programs and services. Business parameters and operating environments vary substantially from customer to customer and as such not all factors, which may impact sizing, have been accounted for in this documentation.

Table 3–3 provides the minimum hardware requirements for Indexing and Search Service.

Table 3–3 Minimum Hardware Requirements

Component	**Requirement**
Disk Space	Approximately 20 MB required for Indexing and Search Service software; must be free in **/opt** for **/opt/sun/comms/jiss**. Roughly 20 percent of the Messaging Server message store size is required for the Indexing and Search Service index store, which by default is in the **/var/opt/sun/comms/jiss** directory.
RAM	16 GB

In Indexing and Search Service, the **/var/opt/sun/comms/jiss** directory is divided into three subdirectories:

- **/var/opt/sun/comms/jiss/attach** - Stores generated thumbnail images from attachments, which tends to consume 2 to 3 percent of the size of the mail store. Turning off the attachment store is a configurable option. Compared to the index store, this directory could probably reside on a slower disk.
- **/var/opt/sun/comms/jiss/logs** - Stores log files.
- **/var/opt/sun/comms/jiss/index** - Stores index files. This tends to take up about 15 percent of the size of the mail store and must be on a fast disk because it is frequently accessed.

Information Requirements

During the Indexing and Search Service installation, you must enter values for configuration items such as host names and port numbers. This section describes the information that you must provide during the installation and initial configuration process.

Indexing and Search Service Cluster Information

Table 3–4 lists the Indexing and Search Service cluster information that you provide during initial configuration.

Table 3–4 Indexing and Search Service Cluster Information

Information Type	Default Value	Comments
Cluster configuration	**standalone**	Choices are: ■ **standalone**: All Indexing and Search Service components are installed on this host. ■ **multimachine**: Some Indexing and Search Service components are installed on this host, which is part of a multiple host deployment. ■ **cluster**: Indexing and Search Service cluster search is enabled. ■ **clusterv2**: Indexing and Search Service cluster search with enhanced firewall security is enabled.
Node type	None	If you choose **multimachine**, **cluster**, or **clusterv2** for cluster configuration, the following node types are available: ■ **all**: Configure all Indexing and Search Service components. ■ **web**: Configure GlassFish Server. ■ **jmq**: Configure Message Queue. ■ **ldap**: Configure Directory Server. ■ **cluster**: Configure Cluster Services. ■ **index**: Configure Indexing Services. ■ **watcher**: Configure Watcher Service.

Indexing and Search Service Information

Table 3–5 lists the Indexing and Search Service information that you provide during initial configuration.

Table 3–5 Indexing and Search Service Information

Information Type	Default Value
Host name of this system	FQDN of host
Instance name of the installation for an indexing node	None
Run-time user ID under which Indexing and Search Service runs	**jiss**
Run-time group to contain Indexing and Search Service run-time user ID	**jiss**
Location to store the Lucene indexes	**/var/opt/sun/comms/jiss/index**
Location of attachment data	**/var/opt/sun/comms/jiss/attach**
Location of Indexing and Search Service log files	**/var/opt/sun/comms/jiss/logs**
Location of Java	**/usr/jdk/latest**

Messaging Server Message Store Information

Table 3–6 lists the message store information that you provide during initial configuration.

Table 3–6 Message Store Information

Information Type	Default Value	Comments
User/Group Directory Server	None	Use a comma-delimited list of Directory Server hosts, for example: **host1.example.com:389,host2.example.com:389**
User/Group Default Domain	None	No comments
User/Group Directory Manager DN	**cn=Directory manager**	No comments
Directory manager password	None	No comments
Messaging Server store host name	None	Specify the FQDN of the default Messaging Server host that is indexed.
IP addresses corresponding to Message Server message store host	None	Use a comma-delimited list of IP addresses corresponding to the Messaging Server store host.
Read-only message store administrative user	**indexeradmin**	This user must exist in the Directory Server LDAP and must be a member of the Messaging Server **store.admins** group.
Read-only message store administrative password	None	No comments
Messaging Server IMAP port	**143**	No comments
Messaging Server JMQ broker host	None	The format is *host***:***port*, for example: **jmq1.example.com:7676**.
JMQ notification user	**jmquser**	The specified user must match the user specified by the Messaging Server **local.store.notifyplugin.index.jmquser** configuration parameter.
JMQ notification user's password	None	The specified password must match the password specified by the Messaging Server **local.store.notifyplugin.index.jmqpwd** configuration parameter.

GlassFish Server Information

Table 3–7 lists the GlassFish Server information that you provide during initial configuration.

Table 3–7 GlassFish Server Information

Information Type	Default Value
GlassFish Server installation directory	**/opt/glassfish3/glassfish**
GlassFish Server web port	**8080**
GlassFish Server administration server port	**4848**
GlassFish Server domain	**domain1**

Table 3–7 (Cont.) GlassFish Server Information

Information Type	Default Value
GlassFish Server administrator user	**admin**
GlassFish Server administrator user password	No default value.
Deploy the SearchUI web application	**false**
Enable the **iss.user** to manage services by using the **svcadm** command	**true**

Message Queue Information

Table 3–8 lists the Message Queue information that you provide during initial configuration.

Table 3–8 JMQ Information

Information Type	Default Value	Comments
Indexing and Search Service JMQ broker host	None	Use a comma-delimited list of JMQ broker hosts, for example: **jmq1.example.com:7676,jmq2.example.com:7676**.
Indexing and Search Service JMQ broker user	**jmquser**	No comments
Indexing and Search Service JMQ broker user's password	None	No comments
Indexing and Search Service JMQ broker administrative user's password	None	No comments

Directory Server JNDI Information

Table 3–9 lists the information about connecting to directory services that hold the Java Naming and Directory Interface (JNDI) lookup that you provide during initial configuration for clustering and multiple host deployments.

Table 3–9 JNDI Information

Information Type	Default Value	Comments
Directory Server host enabled	**false**	In a multiple host deployment, defines the system on which the LDAP configuration is applied. With all Indexing and Search components deployed on a single host, set to **`true`**. Because Indexing and Search Service has LDAP configuration data that must be applied to the LDAP host, this parameter controls which Indexing and Search Service host applies that data. This parameter does not control whether a local LDAP instance is running on the Indexing and Search Service host.
File-based or Directory Server-based JNDI lookups	None	Values are **file** or **ldap**. Use **file** for a standalone installation, and **ldap** for multiple host or cluster installations. Depending on how you have previously answered questions, this information type might not appear.
Directory Server host and port	None	Use a comma-delimited list of Directory Server hosts, for example: **ldap1.example.com:389,ldap2.example.com:389**.
Indexing and Search Service Directory Manager DN	**cn=Directory Manager**	No comments
Indexing and Search Service Directory Server password	None	No comments

Run-time Information

Table 3–10 lists the run-time configurations for Indexing and Search Service services that you provide during initial configuration.

Table 3–10 Run-time Information

Information Type	Default Value	Comments
Storeui access method	**disk**	Specify either **`disk`** for a single host environment or **`http`** for a multiple host environment.

Watcher Service Information

Table 3–11 lists the watcher service information for Indexing and Search Service services that you provide during initial configuration.

Table 3–11 Watcher Service Information

Information Type	Default Value
Enables watcher service	**false**

ishttpd Information

Table 3–12 lists the Web Services Proxy (**ishttpd**) information for Indexing and Search Service services that you provide during initial configuration.

Table 3–12 ishttpd Information

Information Type	Default Value
Enable **ishttpd** service	**false**

4

Indexing and Search Service Pre-Installation Tasks

This chapter describes the pre-installation tasks that you must complete before you can install Oracle Communications Indexing and Search Service.

Installing Java

Oracle GlassFish Server is a Java application and needs a Java environment in which to run.

The 32-bit and 64-bit JDKs require manual installation. Install both JDKs, rather than the JRE, on your front-end hosts.

To get the Java software, go to the Java SE Downloads at:

`http://www.oracle.com/technetwork/java/javase/downloads/index.html`

To install and configure Java, see the Java documentation at:

`http://docs.oracle.com/javase/7/docs/webnotes/install/`

Installing GlassFish Server

To install and configure GlassFish Server, see *Oracle GlassFish Server 3.1.2 Installation Guide* at:

`http://docs.oracle.com/cd/E26576_01/doc.312/e24935/installing.htm#ggssq`

Installing Messaging Server and Directory Server

Indexing and Search Service requires that Oracle Communications Messaging Server and Oracle Directory Server Enterprise Edition be installed. As part of the Directory Server installation, you prepare the Directory Server LDAP schema by running the **comm_dssetup.pl** script. This script, which is provided as part of the Messaging Server software, adds the necessary schema to the LDAP. For more information about installing Messaging Server, and running the **comm_dssetup.pl** script, see *Messaging Server Installation and Configuration Guide*.

Installing Apache HTTP Server Version 2

When you install Indexing and Search Service in a multiple host deployment, where the web tier is running on separate systems from the indexing tier, you must install and configure Apache HTTP Server version 2 on the Indexing and Search Service back-end hosts. That is, you need Apache installed on the hosts that offer the Indexing Tier services (**searchSvc** for searching; **indexSvc** for adding new users and processing mail server events; and **jmqconsumer** for listening to the Messaging Server JMQ for events to process and send to **indexSvc**).

The following software is required:

- Solaris: Ensure that you have installed the package metacluster **SUNWCapache** from your Solaris media. For more information, see the Solaris product documentation.
- Linux: You must install the RPM **httpd** from the Linux media. For instructions, refer to the Linux product documentation.

Installing Message Queue

Indexing and Search Service uses Message Queue for notification services. You can either install an embedded Message Queue that comes with GlassFish Server, or you can install a standalone Message Queue from the Indexing and Search Service software. After installing the Message Queue, you configure it as described in "Configuring an Embedded Message Queue on the Indexing and Search Service Host".

> **Note:** Using the Message Queue that comes with GlassFish Server ensures that you are using the most current version of software.

Installing a Standalone Message Queue from the Indexing and Search Service Software

To install Message Queue from the Indexing and Search Service software on a host:

1. If necessary, download the Indexing and Search Service software (which also contains the Message Queue software) for your operating system from the Oracle software delivery website, located at:

   `http://edelivery.oracle.com/`

2. Copy the Indexing and Search Service ZIP file to a temporary directory on your hosts and extract the files.
3. Change to the directory where you extracted the files.
4. Run the following command to install just Messaging Queue:

   ```
   commpkg install --comp=JISS
   ```

This command installs only the components, such as Message Queue, on which Indexing and Search Service has a dependency, not the Indexing and Search Service software itself.

Installing an Embedded Message Queue from GlassFish Server Software

To install an embedded Message Queue from the GlassFish Server software on a host:

1. If necessary, download the GlassFish software for your operating system from the My Oracle Support website, located at:

   `http://support.oracle.com/`

2. Copy the Glassfish Server ZIP file to a temporary directory on your hosts and extract the files.
3. Install the GlassFish Server software.

For example, to do a silent install of the GlassFish Server ZIP file on a Solaris host:

```
ogs-3.1.2-14-b01-unix-ml.sh -s
```

You can also use a statefile for the installation if you choose.

Running Network Time Protocol

On both the Messaging Server and Indexing and Search Service hosts, ensure that the Network Time Protocol (NTP) service is running.

On Solaris, run the following commands:

```
svcadm enable ntp
svcs ntp
```

For more information about NTP on Solaris, see the topic on time-related services in *System Administration Guide: Network Services*.

On Linux, run the following commands:

```
cd /etc/rc3.d
ln -s ../init.d/ntpd S74ntpd
./init.d/ntpd start
Starting ntpd:                                            [  OK  ]
ps -ef | grep ntp
ntp       2347     1  0 20:56 ?        00:00:00 ntpd -u ntp:ntp -p
/var/run/ntpd.pid -g
```

For more information about NTP on Red Hat Linux, see the section about NTP in Red Hat Documentation.

Increasing the Number of File Handles and Processes

For Linux, increase the number of file handles and number of processes allowed for the **iss.user**.

To increase the number of file handles and processes:

1. As **root**, edit the **/etc/security/limits.conf** file and add the following lines, where **jiss** is the **iss.user** in the **jiss.conf** file:

   ```
   jiss        hard    nofile          65536
   jiss        hard    nproc           8192
   jiss        soft    nofile          65536
   jiss        soft    nproc           8192
   ```

2. As **root**, edit the **/etc/security/limits.d/90-nproc.conf** file and add the following line:

   ```
   * soft nproc 8192
   ```

5

Installing Indexing and Search Service

This chapter describes how to install and configure Oracle Communications Indexing and Search Service. If you are installing Indexing and Search Service in a highly available deployment, follow the instructions in "Configuring Indexing and Search Service for High Availability".

Before installing Indexing and Search Service, read these chapters:

- Indexing and Search Service Installation Overview
- Planning Your Indexing and Search Service Installation
- Indexing and Search Service System Requirements
- Indexing and Search Service Pre-Installation Tasks

Installation Assumptions

The instructions in this chapter assume that:

- You are deploying Indexing and Search Service on a single host or Solaris zone, or multiple hosts or Solaris zones.
- Oracle Communications Messaging Server is already installed and prepared for use by Indexing and Search Service.
- Oracle Directory Server Enterprise Edition (Directory Server) is already installed.
- You have installed, configured, and started Oracle GlassFish Server as the web container for Indexing and Search Service.

Installing Indexing and Search Service

The tasks to install Indexing and Search Service are as follows:

- Downloading the Indexing and Search Service Software
- Installing the Indexing and Search Service Software

Downloading the Indexing and Search Service Software

1. Download the Indexing and Search Service software from the Oracle software delivery website, located at:

   `http://edelivery.oracle.com/`

2. Copy the Indexing and Search Service distribution ZIP file to a temporary directory on your Indexing and Search Service hosts and extract the files.

Installing the Indexing and Search Service Software

Choose one of the following methods to install Indexing and Search Service software:

- Installing Indexing and Search Service Software in Interactive Mode
- Installing Indexing and Search Service in Silent Mode

Installing Indexing and Search Service Software in Interactive Mode

To install the Indexing and Search Service software in interactive mode:

1. Go to the directory where you extracted the Indexing and Search Service files.
2. Run the Installer.

   ```
   commpkg install
   ```

   For more information about running the Installer, see "commpkg Reference".
3. Select **Indexing and Search Service** and proceed with the installation.
4. Proceed to "Preparing Messaging Server for Indexing and Search Service Integration".

Installing Indexing and Search Service in Silent Mode

When you run the installer in silent mode, you are running a non-interactive session. The installation inputs are taken from the following sources:

- A silent installation file (also known as a state file)
- Command-line arguments
- Default settings

You can use silent mode to install multiple instances of the same software component and configuration without having to manually run an interactive installation for each instance.

To run a silent installation:

1. Obtain the state file by one of the following means.
   - Use a state file that was previously created from an interactive installation session. State files are created in the **/var/opt/CommsInstaller/logs/** directory. The state file name is similar to **silent_CommsInstaller_20070501135358**. A state file is automatically created every time you run the **commpkg install** command.
   - Create a silent state file without actually installing the software during the interactive session by running the **commpkg install --dry-run** command.
2. Copy the state file to each host and modify the file as needed.

   The state file is formatted like a property file: blank lines are ignored, comment lines begin with a number sign (#), and properties are key/value pairs separated by an equals (=) sign. Table 5–1 lists the state file options.

Table 5–1 State File Options

Option	Description	Example
VERB	Specifies which function to perform. For a silent install, **VERB** is set to **install**.	**VERB=install**
ALTDISTROPATH	Specifies an alternate distro path.	**ALTDISTROPATH=SunOS5.10_ i86pc_DBG.OBJ/release**
PKGOVERWRITE	Specifies a boolean indicating whether to overwrite the existing installation packages.	**PKGOVERWRITE=YES**
INSTALLROOT	Specifies the installation root.	**INSTALLROOT=/opt/sun/comms**
ALTROOT	Specifies a boolean indicating whether to use an alternate root install.	**ALTROOT=yes**
EXCLUDEOS	Specifies to not upgrade operating system patches.	**EXCLUDEOS=YES**
EXCLUDESC	Specifies to exclude shared component patches.	**EXCLUDESC=no**
COMPONENTS	A space separated list of mnemonics of the components to be installed. You can precede the mnemonic with a ~ to indicate that only the shared components for that product be installed.	To specify Indexing and Search Service: **COMPONENTS=JISS** To view a list of mnemonic product names, run the **commpkg info --listPackages** command.
ACCEPTLICENSE	This option is no longer used.	NA
UPGRADESC	Specifies whether to upgrade all shared components without prompting.	**UPGRADESC=no**
INSTALLNAME	The friendly name for the **INSTALLROOT**.	**INSTALLNAME=**
COMPONENT_ VERSIONS	This option is unused.	NA

3. Run the silent installation on each host.

   ```
   commpkg install --silent input_file
   ```

   where *input_file* is the path and name of the silent state file, for example **/var/opt/CommsInstaller/logs/silent_CommsInstaller_20070501135358**.

   For more information about the **--silent** option, see "install Verb Syntax".

   Note: Command-line arguments override the values and arguments in the state file.

4. Proceed to "Preparing Messaging Server for Indexing and Search Service Integration".

Upgrading Shared Components in Silent Installation

By default, shared components that require user acceptance for upgrading are not upgraded when you run a silent installation. The option to upgrade shared components in the state file is automatically disabled (the **UPGRADESC** option is set to **No**.) This is true even if you explicitly asked to upgrade shared components when you ran the interactive installation that generated the state file. That is, you ran either **commpkg install --upgradeSC y** or you answered “yes” when prompted for each shared component that needed upgrading.

Disabling upgrading shared components in the silent state file is done because the other hosts on which you are propagating the installation might have different shared components installed, or different versions of the shared components. Therefore, it is safer to not upgrade the shared components by default.

You can upgrade shared components when you run a silent installation by performing either of the following actions:

- Use the **--upgradeSC y** option when you run the silent installation. (The command-line argument overrides the argument in the state file.)
- Edit the **UPGRADESC=No** option in the silent state file to: **UPGRADESC=Yes**.

> **Caution:** If you do not upgrade shared components your installation might not work properly.

Installing Indexing and Search Service on Solaris Zones

This section explains how to install Indexing and Search Service on Solaris Zones.

Installing on Solaris Zones: Best Practices

You can install Indexing and Search Service components in the global zone, whole root non-global zones, and sparse non-global zones. Follow these guidelines:

- Treat the global zone as an “administration zone.”

  Install shared components and OS patches in the global zone that are to be shared among all zones. However, do not install and run products from the global zone.
- Use whole root non-global zones to run Indexing and Search Service.

  Do not use the global zone or sparse zones. A whole root zone can have versions that are different from other whole root zones, thus giving it a measure of being “self-contained.”

Be aware of the following zone aspects:

- You can have different shared component versions in the whole root non-global zone, but it is not entirely insulated. If you do a packaging or patching operation in the global zone for a shared component, that operation is also attempted in the whole root zone. Thus, to truly have different shared component versions, use an alternate root.
- To avoid affecting whole root zones you can attempt to never install and patch shared components in the global zone. However, it might not be realistic to never have to install or patch a shared component in the global zone. For example, Network Security Services (NSS) is a shared component, but it is part of Solaris. So to expect to never install and patch NSS in the global zone seems unrealistic, especially given it is a security component.
- Although it is not a recommended best practice, you can use Indexing and Search Service in sparse non-global zones. Shared components cannot be installed into the default root because many of them install into the read-only shared file system (**/usr**). Thus, you must run the installer in the global zone to install shared components into the default root. Prepend your selection with ~ in the global zone to install only the dependencies (that is, shared components). You do not have to install in the global zone first before installing in the sparse zone. The installer enables you to continue even when you do not install all the dependencies.

However, upgrading the shared components in the global zone affects the sparse non-global zones, thus requiring downtime for all affected zones simultaneously.

Note: Sparse root zones are not available beginning with Oracle Solaris 11.

Installing into a Non-Global Whole Root Zone

The non-global whole root zone scenario is the equivalent of installing Indexing and Search Service on a single host with no zones. Simply install Indexing and Search Service as described in "Installing Indexing and Search Service".

Caution: Any operations performed in the global zone (such as installations, uninstallations, and patching) affect the whole root zones.

Installing into a Non-Global Sparse Root Zone

Although it is not a recommended best practice, you can use Indexing and Search Service in a non-global sparse root zone on Solaris 10. To install Indexing and Search Service in a non-global sparse root zone, you first must install or upgrade the applicable operating system patches and shared components in the global zone. You are unable to do so in the sparse root zone, because the **/usr** directory (where the shared components reside) is a read-only directory in the sparse root zone.

1. Follow the pre-installation requirements as described in "Indexing and Search Service Pre-Installation Tasks".
2. Verify that you are about to install the shared components and operating system patches in the global zone and not the sparse root zone. To verify that you are in the global zone, run **zonename**. The output should be global.
3. Run the installer in the global zone and only install or upgrade the operating system patches and the Shared Components. Do not install Indexing and Search Service components in the global zone. To install only the Shared Components in the global zone, add a ~ (tilde) to the component number you want to install in the sparse zone.

   For example, if you plan to install Indexing and Search Service in the sparse zone, you select **~1** during the global zone installation. The installer knows to only install dependencies and not the product itself.
4. Once you have installed the shared components and operating system patches, install Indexing and Search Service components in the sparse root zone as described in "Installing Indexing and Search Service".

Configuring an Embedded Message Queue on the Indexing and Search Service Host

To configure an embedded Message Queue on the Indexing and Search Service host:

1. Ensure that the Message Queue broker is running and is enabled to start at reboot. See the topic on updating the JMS service configuration in *Oracle GlassFish Server 3.1 Administration Guide*.

2. If the **config.properties** file does not yet exist, it is generated the first time that the IMQ broker is started. You might need to restart GlassFish Server to cause the file to be generated. Some later versions of GlassFish Server require you to connect to the port before this file is generated, for example:

   ```
   telnet 0 7676
   ```

3. Tune the Message Queue broker properties by editing the bottom of the *GlassFish_home*/**glassfish/domains/domain1/imq/instances/imqbroker/props/config.properties** file to contain the following information.

   ```
   imq.portmapper.backlog=-1
   imq.destination.DMQ.truncateBody=true
   imq.autocreate.reaptime=7200
   imq.log.level=WARNING
   imq.autocreate.destination.maxNumProducers=-1
   imq.jms.max_threads=2000
   ```

4. Restart GlassFish Server to acquire the new settings:

   ```
   GlassFish_home/bin/asadmin stop-domain domain1
   GlassFish_home/bin/asadmin start-domain domain1
   ```

5. Edit the *IndexingSearch_home*/**etc/jiss.conf file**, or a state file, to contain the following information.

   ```
   # Full path to the imqcmd imqusermgr binaries
   iss.imq.bin = GlassFish_home/mq/bin
   #Full path to the imq broker var directory (IMQ_DEFAULT_VARHOME)
   iss.imq.var = GlassFish_home/glassfish/domains/domain1/imq
   ```

6. Run the **setup** command:

   ```
   IndexingSearch_home/bin/setup [-c statefile]
   ```

7. Restart GlassFish Server.

8. Verify the broker:

   ```
   GlassFish_home/mq/bin/imqcmd list dst
   Listing all the destinations on the broker specified by:

   -------------------------
   Host         Primary Port
   -------------------------
   localhost    7676

   ------------------------------------------------------------------------------
   ---------------
      Name      Type    State      Producers        Consumers
   Msgs
                                Total  Wildcard  Total  Wildcard  Count  Remote
   UnAck  Avg Size
   ------------------------------------------------------------------------------
   ---------------
   AccountState.sca00agq  Topic  RUNNING  0      0        3      2        0
   0       0      0.0
   Indexsca00agq          Queue  RUNNING  128    -        1      -        0
   0       0      0.0
   Jmqcsca00agq           Topic  RUNNING  0      0        1      0        0
   0       0      0.0
   SearchTopic            Topic  RUNNING  512    0        1      0        0
   ```

```
0       0      0.0
mq.sys.dmq             Queue  RUNNING  0      -        0      -        0
0       0      0.0

Successfully listed destinations.
```

9. Add Oracle Solaris Service Management Facility (SMF) dependencies.

   When installing Indexing and Search Service with an embedded Message Queue, the SMF definitions for the Indexing and Search Service services are managed by the Indexing and Search Service **setup** utility. Consequently, Indexing and Search Service services do no start unless the GlassFish Message Queue broker is already running. Thus, ensure that GlassFish Server is running prior to starting Indexing and Search Service services.

   You can add this dependency to the SMF service definitions. For example:

```
<dependency name='nab' grouping='optional_all' restart_on='none'
type='service' >
     <service_fmri value='svc:/application/glassfish3/domain1:default'/>
</dependency>
```

Configuring a Standalone Message Queue on the Indexing and Search Service Host

To configure a standalone Message Queue on the Indexing and Search Service host:

1. Ensure that the Message Queue broker is running and is enabled to start at reboot. Edit the **imqbrokerd.conf** file (**/etc/imq/imqbrokerd.conf** on Solaris and **/etc/opt/sun/mq/imqbrokerd.conf** on Linux) to contain the following information. Check the file permissions and change them to be writable if they are set to read-only.

```
AUTOSTART=YES
ARGS=-vmargs -Xmx1024m -vmargs -Xss256k -vmargs -d64
RESTART=YES
```

2. If the **config.properties** file (**/var/imq/instances/imqbroker/props/config.properties** on Solaris and **/var/opt/sun/mq/instances/imqbroker/props/config.properties** on Linux) does not yet exist, start the IMQ broker for the first time to generate it.

```
nohup /etc/init.d/imq start
```

3. Tune the Message Queue broker properties by editing the bottom of the **config.properties** file (**/var/imq/instances/imqbroker/props/config.properties** on Solaris and **/var/opt/sun/mq/instances/imqbroker/props/config.properties** on Linux) to contain the following information.

   You might need to check the file permissions and change them to be writable if set to read-only. The defaults for these settings appear in the **default.properties** file (**/usr/share/lib/imq/props/broker/default.properties** on Solaris and **/opt/sun/mq/private/share/lib/props/broker/default.properties** on Linux).

```
imq.portmapper.backlog=-1
imq.autocreate.destination.limitBehavior=REMOVE_OLDEST
imq.autocreate.destination.maxNumProducers=-1
imq.autocreate.reaptime=7200
imq.destination.DMQ.truncateBody=true
imq.jms.max_threads=2000
```

4. Ensure that the Message Queue broker is restarted to acquire the new settings:

   ```
   /etc/init.d/imq stop
   nohup /etc/init.d/imq start
   ```

Preparing Messaging Server for Indexing and Search Service Integration

Preparing your Messaging Server hosts for Indexing and Search Service integration includes:

- Assigning read-only store administrator privileges to the Indexing and Search Service proxy authentication user
- Configuring Java Messaging Queue (JMQ) notifications
- Configuring Messaging Server to forward some IMAP SEARCH queries to Indexing and Search Service

To prepare and configure Messaging Server for integration with Indexing and Search Service:

1. Assign read-only store administrator privileges to the Indexing and Search Service proxy authentication user (by default named **indexeradmin**).

   The Indexing and Search Service proxy authentication user must exist in LDAP and must be a member of the **store.admins** group.

   - Legacy configuration:

     ```
     cd MessagingServer_home/bin
     configutil -o store.indexeradmins -v indexeradmin
     ```

   - Unified Configuration:

     ```
     cd MessagingServer_home/bin
     msconfig set store.indexeradmins indexeradmin
     ```

2. Configure a Message Queue user.

   a. Use the **imqusermgr** command to create a Message Queue user, referred to later in this task as **local.store.notifyplugin.index.jmquser** in Messaging Server and **mail.imq.user** in Indexing and Search Service.

   – Solaris:

   ```
   /usr/bin/imqusermgr add -u jmquser -p jmquserpassword -g user
   ```

   – Linux:

   ```
   /opt/sun/mq/bin/imqusermgr add -u jmquser -p jmquserpassword -g user
   ```

   b. Reset the default account passwords, if not already reset:

   – Solaris:

   ```
   /usr/bin/imqusermgr update -u admin -p adminpassword
   /usr/bin/imqusermgr update -u guest -p guestpassword
   ```

   – Linux:

   ```
   /opt/sun/mq/bin/imqusermgr update -u admin -p adminpassword
   /opt/sun/mq/bin/imqusermgr update -u guest -p guestpassword
   ```

3. Configure Java Messaging Queue (JMQ) notifications on Messaging Server to produce event notifications of real-time changes to the Messaging Server store for Indexing and Search Service to consume.

> **Note:** Indexing and Search Service requires JMQ software. If on your Messaging Server hosts you are currently running ENS for IMAP IDLE, you must run both ENS and JMQ. In other words, ENS and JMQ are independent. You can use ENS for IMAP IDLE and JMQ for Indexing and Search Service. Prior to Messaging Server 7 Update 4, you can use JMQ for both IMAP IDLE and Indexing and Search Service. Starting with Messaging Server 7 Update 4, IMAP IDLE only supports ENS.

a. Set the **local.store.notifyplugin** parameter (legacy configuration) or **notifytarget** option (Unified Configuration) to include a JMQ notification plug-in named **index**.

If you have already enabled **local.store.notifyplugin**, add the following entry for Indexing and Search Service to what is already specified so that you do not overwrite the existing setting. For example, if **local.store.notifyplugin** is already set, run the following commands.

– Legacy configuration:

```
cd MessagingServer_home/bin
configutil -o local.store.notifyplugin
lib/libibiff$ms-internal$
configutil -o local.store.notifyplugin -v
'lib/libibiff$ms-internal$lib/libjmqnotify$index'
```

You may receive a response such as the following:

```
libibiff$ms-internal$
```

If this is the case, append the new JMQ notification to existing one:

```
configutil -o local.store.notifyplugin -v
'libibiff$ms-internal$libjmqnotify$index'
```

If **local.store.notifyplugin** is not already set, set it as follows:

```
cd MessagingServer_home/bin
configutil -o local.store.notifyplugin
configutil -o local.store.notifyplugin -v 'lib/libjmqnotify$index'
```

– Unified Configuration:

```
msconfig set notifytarget:index.notifytype jmq
```

For more information about the syntax of **local.store.notifyplugin**, consult *Messaging Server Reference* or run the **configutil -o local.store.notifyplugin -H** command.

b. Configure the following settings for the index JMQ notification plugin, replacing *jmquser* with the JMQ user and *jmquserpassword* with the JMQ user password. Time values are in milliseconds.

– Legacy configuration:

```
cd MessagingServer_home/bin
configutil -o local.store.notifyplugin.index.annotatemsg.enable -v 0
configutil -o local.store.notifyplugin.index.changeflag.enable -v 1
configutil -o local.store.notifyplugin.index.copymsg.enable -v 1
configutil -o local.store.notifyplugin.index.debuglevel -v 0
configutil -o local.store.notifyplugin.index.deletemsg.enable -v 0
configutil -o local.store.notifyplugin.index.destinationtype -v queue
configutil -o local.store.notifyplugin.index.expungemsg.enable -v 1
configutil -o local.store.notifyplugin.index.jmqhost -v ISS_host_FQDN
configutil -o local.store.notifyplugin.index.jmqport -v 7676
configutil -o local.store.notifyplugin.index.jmqpwd -v jmquserpassword
configutil -o local.store.notifyplugin.index.jmqtopic -v INDEXMS
configutil -o local.store.notifyplugin.index.jmquser -v jmquser
configutil -o local.store.notifyplugin.index.loguser.enable -v 0
configutil -o local.store.notifyplugin.index.maxbodysize -v 262144
configutil -o local.store.notifyplugin.index.maxheadersize -v 16384
configutil -o local.store.notifyplugin.index.msgflags.enable -v 0
configutil -o local.store.notifyplugin.index.msgtypes.enable -v 0
configutil -o local.store.notifyplugin.index.newmsg.enable -v 1
configutil -o local.store.notifyplugin.index.noneinbox.enable -v 1
configutil -o local.store.notifyplugin.index.overquota.enable -v 0
configutil -o local.store.notifyplugin.index.persistent -v 0
configutil -o local.store.notifyplugin.index.priority -v 4
configutil -o local.store.notifyplugin.index.purgemsg.enable -v 0
configutil -o local.store.notifyplugin.index.readmsg.enable -v 0
configutil -o local.store.notifyplugin.index.underquota.enable -v 0
configutil -o local.store.notifyplugin.index.updatemsg.enable -v 1
configutil -o local.store.notifyplugin.index.ttl -v 3600000
```

– Unified Configuration:

```
msconfig set notifytarget:index.annotatemsg 0
msconfig set notifytarget:index.changeflag 1
msconfig set notifytarget:index.copymsg 1
msconfig set notifytarget:index.deletemsg 0
msconfig set notifytarget:index.destinationtype queue
msconfig set notifytarget:index.expungemsg 1
msconfig set notifytarget:index.jmqhost ISS_host_FQDN
msconfig set notifytarget:index.jmqport 7676
msconfig
## The JMQ password can only be set using the msconfig interactive
mode.
msconfig> set notifytarget:index.jmqpwd jmquserpassword
msconfig# write
msconfig set notifytarget:index.jmqtopic INDEXMS
msconfig set notifytarget:index.jmquser jmquser
msconfig set notifytarget:index.loguser 0
msconfig set notifytarget:index.maxbodysize 262144
msconfig set notifytarget:index.maxheadersize 16384
msconfig set notifytarget:index.msgflags 0
msconfig set notifytarget:index.msgtypes 0
msconfig set notifytarget:index.newmsg 1
msconfig set notifytarget:index.noninbox 1
msconfig set notifytarget:index.overquota 0
msconfig set notifytarget:index.persistent 0
msconfig set notifytarget:index.priority 4
msconfig set notifytarget:index.purgemsg 0
msconfig set notifytarget:index.readmsg 0
msconfig set notifytarget:index.underquota 0
msconfig set notifytarget:index.updatemsg 1
msconfig set notifytarget:index.ttl 3600000
```

If you are deploying multiple Indexing and Search Service back- end hosts to connect to the same **INDEXMS** destination, set the **local.store.notifyplugin.index.destinationtype** parameter (legacy configuration), or **notifytarget:index.destinationtype** option (Unified Configuration) to **`topic`** rather than **`queue`**.

The **local.store.notifyplugin.index.overquota.enable** and **local.store.notifyplugin.index.underquota.enable** parameters (legacy configuration), or **notifytarget:index.overquota** and **notifytarget:index.underquota** (Unified Configuration) are "newer" parameters that older versions of the **configutil** command do not support. If this is the case, you see a warning about setting an unknown parameter. This only means that the over and under quota events have not been disabled. These events are still sent to Indexing and Search Service but are ignored.

c. Important: For Messaging Server 7 Update 3 or earlier versions, if you have changed the guest password for the IMQ broker from its default value and you do not have ENS enabled for IDLE, you might also need to either disable IDLE and **imkill** over JMQ, or provide a valid JMQ user name and password for their configuration.

- To disable IDLE and KILL over JMQ, run the following commands:

```
cd MessagingServer_home/bin
configutil -o service.imap.ensidle -v 1
```

This setting does not enable ENS. Instead, it disables IDLE and **imkill** over JMQ.

- To provide a valid JMQ user name and password for IDLE and JMQ, run the following commands:

```
cd MessagingServer_home/bin
configutil -o service.imap.idle.jmquserid -v jmquser
configutil -o service.imap.idle.jmqpassword -v jmquserpassword
configutil -o local.store.notifyplugin.idle.jmquser -v jmquser
configutil -o local.store.notifyplugin.idle.jmqpwd -v jmquserpassword
```

Starting with Messaging Server 7 Update 4, the **service.imap.idle.jmquserid** and **service.imap.idle.jmqpassword** parameters might be obsolete and might send a warning message if you attempt to set them. Beginning with Messaging Server 7 Update 5, run these commands instead:

```
cd MessagingServer_home/bin
configutil -o local.store.notifyplugin.idle.jmquser -v jmquser
configutil -o local.store.notifyplugin.idle.jmqpwd -v jmquserpassword
```

- If you skip this step, but the following messages appear in the IMAP log after you restart Messaging Server, you must perform one of the previous two options.

```
[09/Apr/2009:22:59:05 +0000] mailhost imapd[26954]: General Error: JMQ
Error: Broker: invalid login
[09/Apr/2009:22:59:16 +0000] mailhost imapd[26954]: General Error: JMQ
Error: Broker: invalid login
[09/Apr/2009:22:59:26 +0000] mailhost imapd[26954]: General Error: JMQ
Error: Broker: invalid login
[09/Apr/2009:22:59:37 +0000] mailhost imapd[26954]: General Error: JMQ
Error: Broker: invalid login
[09/Apr/2009:22:59:47 +0000] mailhost imapd[26954]: General Error: JMQ
Error: Broker: invalid login
```

```
[09/Apr/2009:22:59:58 +0000] mailhost imapd[26954]: General Error: JMQ
Error: Broker connection is closed
```

d. Restart Messaging Server to make the changes take effect:

```
cd MessagingServer_home/bin
stop-msg
start-msg
```

e. Verify that the JMQ index plugin is working by checking for an entry such as the following in the *MessagingServer_home*/**log/imap** log:

```
[28/May/2013:22:35:35 +0000] mailhost imapd[13996]: General Notice: JMQ
notifica tions enabled: index
[28/May/2013:22:35:35 +0000] mailhost imapd[13996]: General Notice:
jmqNotify JM Q initialization MQCreateConnection [dest : index] [host :
127.0.0.1]
[28/May/2013:22:35:35 +0000] mailhost imapd[13996]: General Notice:
jmqNotify JM Q initialization MQCreateSession [dest : index] [host :
127.0.0.1]
[28/May/2013:22:35:35 +0000] mailhost imapd[13996]: General Notice:
jmqNotify JM Q initialization MQCreateDestination [dest : index] [host :
127.0.0.1] [topic :  INDEXMS]
[28/May/2013:22:35:35 +0000] mailhost imapd[13996]: General Notice:
jmqNotify JM Q initialization MQCreateMessageProducerForDestination [dest
: index] [host : 12 7.0.0.1]
[28/May/2013:22:35:35 +0000] mailhost imapd[13996]: General Notice:
jmqNotify JM Q initialization MQCreateTextMessage [dest : index] [host :
127.0.0.1]
[88/May/2013:15:11:07 -0700] mailhost imapd[19507]: General Warning:
Oracle Communications Messaging Server IMAP4 7u5-28.21(7.0.5.28.0) 64bit
(built Apr 8 2013) starting up
```

Messaging Server 7 Update 4 and later may require that the NSS shared component be updated before JMQ notifications works properly. If the *MessagingServer_home*/**log/imap** log contains an error at start up similar to the following:

```
[26/Jun/2012:18:55:00 +0000] bco44 imapd[5471]: General Error: jmqNotify
JMQ initialization error The library is incompatible
```

Then consult the patch README for Messaging Server to determine the corresponding required patch level for the NSS/NSPR/JSS patch. For example, Messaging Server 7 Update 4 Patch 25 requires the 3.12.8 version of NSS shared component.

f. On the GlassFish Server host, verify that no login errors have occurred on the IMQ broker by checking the **/var/imq/instances/imqbroker/log/log.txt** file. For example:

```
[06/Apr/2009:19:51:32 PDT] [B1065]: Accepting:
jmquser@127.0.0.1:65220->jms:63268. Count: service=1 broker=1
[06/Apr/2009:19:51:33 PDT] [B1065]: Accepting:
jmquser@127.0.0.1:65223->jms:63268. Count: service=2 broker=2
[06/Apr/2009:19:51:34 PDT] [B1065]: Accepting:
jmquser@127.0.0.1:65226->jms:63268. Count: service=3 broker=3
[06/Apr/2009:19:51:34 PDT] [B1065]: Accepting:
jmquser@127.0.0.1:65228->jms:63268. Count: service=4 broker=4
[06/Apr/2009:19:51:35 PDT] [B1065]: Accepting:
jmquser@127.0.0.1:65232->jms:63268. Count: service=5 broker=5
```

g. On the GlassFish Server host, verify that Messaging Server has producers connected to the **INDEXMS** destination:

```
imqcmd list dst
Username: admin
Password:
Listing all the destinations on the broker specified by:

-------------------------
Host         Primary Port
-------------------------
localhost    7676

--------------------------------------------------------------------------
-------------------
   Name      Type    State      Producers       Consumers
Msgs
                             Total  Wildcard  Total  Wildcard  Count
Remote  UnAck  Avg Size
--------------------------------------------------------------------------
-------------------
INDEXMS      Queue  RUNNING  9     -         0     -         0      0
0      0.0
mq.sys.dmq   Queue  RUNNING  0     -         0     -         0      0
0      0.0

Successfully listed destinations.
```

h. For more information, see the topic on JMQ notification in *Messaging Server System Administrator's Guide*.

4. Prepare, but do not enable, Messaging Server to forward some IMAP SEARCH queries to Indexing and Search Service.

   After you have enabled Indexing and Search Service, bootstrapped user accounts, and verified that search works, this step is revisited to enable redirecting of IMAP SEARCH queries to Indexing and Search Service by setting **service.imap.indexer.enable** to **1**. At this point in the installation, enabling IMAP SEARCH queries to be redirected to Indexing and Search Service would result in search errors.

   Indexing and Search Service processes search operations based on words rather than substrings. Consequently, results might not follow IMAP standards and might differ from a search performed by the IMAP server. For more information about Indexing and Search Service search scenarios, see "How Indexing and Search Service Performs Searches".

   If an error occurs from Indexing and Search Service, then the search might fall back to processing by the IMAP server. See *Indexing and Search Service System Administrator's Guide* for details on what searches Indexing and Search Service supports, because it might not support all combinations of AND, OR and NOT operations that IMAP supports.

   a. Before Indexing and Search Service is enabled, apply the following settings:

      – Legacy configuration:

```
cd MessagingServer_home/bin
configutil -o service.imap.indexer.hostname -v glassfish_hostname
configutil -o service.imap.indexer.port -v glassfish_port
configutil -o service.imap.indexer.enable -v 0
```

– Unified Configuration:

```
msconfig set imap.indexer.server_host glassfish_hostname
msconfig set imap.indexer.port glassfish_port
msconfig set imap.indexer.enable 0
```

Where:

glassfish_hostname is the GlassFish Server host upon which Indexing and Search Service web component is configured

glassfish_port is the port number of the GlassFish Server host upon which Indexing and Search Service web component is configured

b. Restart the Messaging Server IMAP server to make the changes take effect:

```
cd MessagingServer_home/bin
refresh imap
```

Configuring Indexing and Search Service

You must configure Indexing and Search Service to complete the installation. You use the Indexing and Search Service configuration command-line script, **setup**, to perform this initial run-time configuration.

The **setup** script generates configuration files required to run the Indexing and Search Service services. The script uses the *IndexSearch_home*/**etc/jiss.conf.template** file for default values but you are prompted for various configuration settings about the Messaging Server messaging store install, local system services, and so on.

Running the Indexing and Search Service Initial Configuration Script

Before you begin running the initial configuration script, ensure that the Directory Server, JMQ broker, GlassFish Server, and Messaging Server components are running.

To run the Indexing and Search Service initial configuration:

1. Log in as or become the superuser (**root**).

2. Change to the *IndexSearch_home*/**bin** directory.

   The default installation directory is **/opt/sun/comms/jiss**.

3. Run the initial configuration script and respond to the prompts. The prompts might change depending on your selections.

   For more information about options to the **setup** script, see "setup Script".

   ```
   setup -b IndexSearch_home -t type
   ```

   Note: Refer to "Information Requirements" for information about the values you must provide during initial configuration.

4. When prompted, enter the Indexing and Search Service cluster settings:

   - Cluster configuration
   - Node type (appears if you chose **multimachine**, **cluster**, or **clusterv2** for cluster configuration)

5. Enter the Indexing and Search Service Settings.

   - Host name of this system
   - Instance name of the installation for an indexing node
   - Run-time user ID under which Indexing and Search Service runs
   - Run-time group to contain Indexing and Search Service run-time user ID
   - Location to store the Lucene indexes
   - Location of attachment data
   - Location of Indexing and Search Service log files
   - Location of Java

6. Enter the Message Server Message Store Settings.
   - User/Group Directory Server
   - User/Group Default Domain
   - User/Group Directory Manager DN
   - Directory manager password
   - Messaging Server store host name
   - IP addresses corresponding to Message Server message store host
   - Read-only message store administrative user
   - Read-only message store administrative password
   - Messaging Server IMAP port
   - Messaging Server JMQ broker host
   - JMQ notification user
   - JMQ notification user's password
7. Enter the GlassFish Server Settings.
   - GlassFish Server installation directory
   - GlassFish Server web port
   - GlassFish Server administration server port
   - GlassFish Server domain
   - GlassFish Server administrator user
   - GlassFish Server administrator user password
   - Deploy the SearchUI web application
   - Enable the **iss.user** to manage services by using the **svcadm** command
8. Enter the Message Queue Settings.
   - Indexing and Search Service JMQ broker host
   - Indexing and Search Service JMQ broker user
   - Indexing and Search Service JMQ broker user’s password
   - Indexing and Search Service JMQ broker administrative user’s password
9. Enter the Directory Server JNDI Settings.
   - Directory Server host enabled

- File-based or Directory Server-based JNDI lookups
- Directory Server host and port (required for multiple machine and cluster deployments)
- Indexing and Search Service Directory Manager DN
- Indexing and Search Service Directory Server password

10. Enter the Run-time Information Settings.
    - Storeui access method
11. Enter the Watcher Information Settings.
12. Enter the ishttpd Information Settings.
13. When prompted whether to proceed with configuring Indexing and Search Service, answer **Y**.

    The configurator displays messages indicating its actions and progress. The last messages indicate the location where the installation log file was written.
14. Restart GlassFish Server.

Troubleshooting Indexing and Search Service Installation Problems

This section provides troubleshooting topics if you experience installation problems.

Troubleshooting the Initial Configuration

If you encounter problems during the initial Indexing and Search Service configuration, you can run the **setup uninstall** command, adjust the initial configuration, and re-attempt the setup. For more information, see "setup Script".

Troubleshooting Configuration Problems

This section describes how to troubleshoot installation problems without using the **setup** script.

To troubleshoot configuration problems:

1. Change to the *IndexSearch_home***/bin** directory.
2. Make a copy of the example configuration file.

   ```
   cp ../etc/jiss.conf.example myfile.conf
   ```
3. Edit the **myfile.conf** with your configuration details.
4. Move the existing configuration files.

   ```
   mv ../etc/jiss.conf ../etc/jiss.conf.backup
   mv ../etc/keystore.jks ../etc/keystore.jks.backup
   mv ../etc/stowg ../etc/stowg.backup
   ```
5. Run the **configure_etc.pl** script to create new **jiss.conf** and **jiss.password** files.

   ```
   configure_etc.pl -c myfile.conf -A
   ```
6. Run the **verify_data.pl** script to check the *IndexSearch_home***/etc/jiss.conf** and *IndexSearch_home***/etc/jiss_passwd.conf** files for all required parameters and formatting.

7. Run the **verify_conf.pl** script to check if needed services are running and can be authenticated with provided credentials.
8. If all these steps passed, run **setup** to configure the system.

   ```
   setup -b basedir
   ```

For more information about troubleshooting your deployment, see the troubleshooting topic in *Indexing and Search Service System Administrator's Guide*.

Next Steps

After configuring Indexing and Search Service, continue with the following chapters:

- Go to "Configuring Indexing and Search Service with Multiple Hosts" if you have multiple hosts in your deployment.
- Go to "Configuring Indexing and Search Service Web Services Proxy" if your deployment uses Oracle Communications Convergence.
- Follow the instructions in "Indexing and Search Service Post-Installation Tasks" to perform post-installation tasks.

6

Configuring Indexing and Search Service with Multiple Hosts

This chapter provides examples that show how to configure Oracle Communications Indexing and Search Service in a multiple host environment. If your deployment includes Oracle Communications Convergence, see "Configuring Indexing and Search Service Web Services Proxy".

Using a Default Configuration File

In a multiple host deployment, all hosts should have the same configuration. Create your configuration before running the **setup** command by using a copy of the *IndexSearch_home*/**etc/jiss.conf.example** file. If you concatenate the **jiss.conf** and a list of passwords, see *IndexSearch_home*/**bin/jks --list** to generate the configuration that you are using on all hosts. For more information, see the topic on Java keystore in *Indexing and Search Service Security Guide*.

iss.storeui.access.method Parameter

In a multiple host deployment, if the GlassFish Server and the index reside on different hosts, you must set up an Apache server on the indexing host to provide access to the attachment store to the GlassFish Server. This Apache server runs under the **iss.user** and **iss.group** credentials. If the **iss.storeui.access.method** parameter is set to **http**, the **setup** command calls **configure_apache.sh**, which configures the Apache server to use **iss.user** and **iss.group** running on the port configured by the **apache.port** parameter.

Order in Which to Install Multiple Hosts

In a multiple host deployment, configure the Indexing and Search Service components by running the **setup** script in the following order.

1. **ldap** - Holds JNDI lookups
2. **ms** - Message store configuration
3. **jmq** - Requires JNDI lookups
4. **index** - Requires JNDI lookup and Message Queue
5. **web** - Requires index services to be started
6. **watcher** - Provides local host monitoring of Indexing and Search Service services

For more information on each component, see "Deciding on Which Indexing and Search Service Components to Install"

When using an Indexing and Search Service host to perform the Directory Server function for JNDI lookups, you run the **setup ldap** command on the Indexing and Search Service host, not on the Directory Server host (which typically serves your entire deployment). The Indexing and Search Service LDAP component enables the external Directory Server to host Indexing and Search Service's JDNI lookups. The Indexing and Search Service host that you use to run the command must have access to the **ldapmodify** and **ldapsearch** commands.

Multiple Hosts Examples

This section provides examples of multiple host deployments and which Indexing and Search Service components to configure on which hosts.

The first time that you run the **setup** script on each machine, you must provide the **-c** *conf_file* and **-i** *fully_qualifed_domain_name* options. Once you have configured the **jiss.conf** and **keystore.jks** files, you can omit these options.

Three Hosts Example

This example describes the following three-host setup:

- **web.example.com** - GlassFish Server (**web**)
- **middle.example.com** - Directory services and Message Queue (**ldap**, **jmq**, and **ms**)
- **index.example.com** - Indexing services (**index**)

To configure this three-host example:

1. Run the following commands on **middle.example.com**:

   ```
   IndexSearch_home/bin/setup -t ldap -c my.conf -i middle.example.com
   IndexSearch_home/bin/setup -t ms
   IndexSearch_home/bin/setup -t jmq
   ```

2. Run the following command on **index.example.com**:

   ```
   IndexSearch_home/bin/setup -t index -c my.conf -i index.example.com
   ```

3. Run the following command on **web.example.com**:

   ```
   IndexSearch_home/bin/setup -t web -c my.conf -i web.example.com
   ```

Two Hosts Example

This example describes the following two-host setup:

- **web.example.com** - GlassFish Server (**web**)
- **index.example.com** - Indexing services, Directory services, and Message Queue (**index**, **ldap**, **jmq**, and **ms**)

To configure this two-host example:

1. Run the following commands on **index.example.com**:

   ```
   IndexSearch_home/bin/setup -t ldap -c my.conf -i index.example.com
   IndexSearch_home/bin/setup -t ms
   IndexSearch_home/bin/setup -t jmq
   IndexSearch_home/bin/setup -t index
   ```

2. Run the following command on **web.example.com**:

   ```
   IndexSearch_home/bin/setup -t web -c my.conf -i web.example.com
   ```

Three Hosts Example Using Two Indexing Hosts

This example describes the following three-host setup:

- **web.example.com** - GlassFish Server (**web**)
- **index1.example.com** - Indexing services, Directory services, and Message Queue (**index**, **ldap**, **jmq**, and **ms**)
- **index2.example.com** - Indexing services (**index**)

To configure this three-host example:

1. Run the following commands on **index1.example.com**:

   ```
   IndexSearch_home/bin/setup -t ldap -c my.conf -i index1.example.com
   IndexSearch_home/bin/setup -t ms
   IndexSearch_home/bin/setup -t jmq
   IndexSearch_home/bin/setup -t index
   ```

2. Run the following command on **index2.example.com**:

   ```
   IndexSearch_home/bin/setup -t index -c my.conf -i index2.example.com
   ```

3. Run the following command on **web.example.com**:

   ```
   IndexSearch_home/bin/setup -t web -c my.conf -i web.example.com
   ```

Performing Additional Configuration on Indexing Hosts

You must configure the Apache server to respond to requests from the web nodes' IP addresses. The default setting is to deny requests. You can add and remove additional hosts by using the **configure_web_node** utility. The IP address(es) should be the IP address(es) of the hosts running GlassFish Server.

To add an IP address:

```
IndexSearch_home/bin/configure_web_node -a ip_address
```

To remove an IP address:

```
IndexSearch_home/bin/configure_web_node -d ip_address
```

To add multiple IP addresses:

```
IndexSearch_home/bin/configure_web_node -a ip_address ip_address
```

Configuring Multiple Messaging Server Message Stores

To index multiple Messaging Server message stores on a single Indexing and Search Service host:

1. Choose one of the message store hosts to be the default mail store for purposes of the Indexing and Search Service installation.

   By setting up a default mail server host name in the index, you do not need to specify it with every **issadmin.sh** command. For example, when the default host name is **ms1.example.com**, then the following two commands are equivalent:

```
issadmin.sh --bootstrap --user user1
issadmin.sh --bootstrap --user user1 --host ms1.example.com
```

For users on the non-default message store host, you do have to specify the host name when running the **issadmin.sh** command, for example:

```
issadmin.sh --bootstrap --user user1 --host ms2.example.com
```

2. For the **mail.server.ip** parameter, which you specify when running the **setup** script, type the IP addresses, separated by commas, for all the mail store hosts.
3. Configure the message store hosts to publish JMQ notifications to a single **INDEXMS** destination by running the Messaging Server **configutil** command to set the following parameters:

```
local.store.notifyplugin.index.jmqhost = jmq_host
local.store.notifyplugin.index.jmqport = jmq_port
local.store.notifyplugin.index.jmqpwd = password
local.store.notifyplugin.index.jmqtopic = INDEXMS
local.store.notifyplugin.index.jmquser = jmquser
```

You can either have all message store hosts publish to the broker running on the Indexing and Search Service host, or choose one of the message store hosts as a JMQ broker location, then have that server connect to localhost to publish notifications, and have the others connect to that host to publish notifications.

7

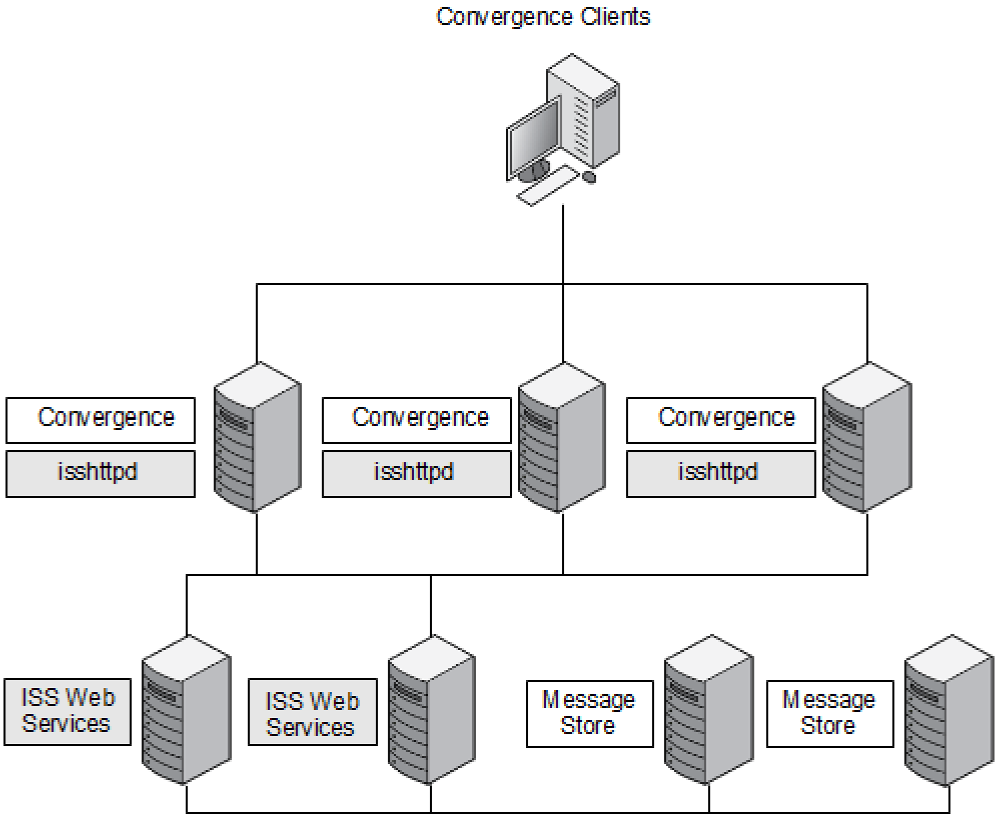
Configuring Indexing and Search Service Web Services Proxy

This chapter describes how to configure the Oracle Communications Indexing and Search Service web services proxy.

Indexing and Search Service Web Services Proxy Overview

Indexing and Search Service provides a Web Services Proxy (**isshttpd**). The Web Services Proxy is a standalone Java daemon that Oracle Communications Convergence utilizes to route search and thumbnail requests to the correct Indexing and Search Service web services host. Figure 7–1 shows a deployment of multiple front-end Convergence hosts using the Indexing and Search Service web services proxy.

Figure 7–1 isshttpd Deployed on Convergence Host



You deploy the **isshttpd** proxy on the same host that runs Convergence. You can use either a flat file lookup or Directory Server LDAP entries to map the user requesting the search, and the user's mailhost, to the user's Indexing and Search Service web services.

The **isshttpd** proxy also provides an API that enables non-Convergence clients, such as mobile clients, to search Indexing and Search Service. No additional configuration is necessary to be able to use this API.

When clients attempt to search Indexing and Search Service, the **isshttpd** proxy first authenticates the user. Next, the proxy constructs a RESTFUL search URL. For Convergence clients, the **isshttpd** proxy generates a search URL based on the user’s **mailHost** attribute. Mobile clients do not have direct knowledge of the user’s email message store. Thus, for non-Convergence clients, the **isshttpd** proxy generates a search URL that contains the user name.

How isshttpd Maps User Entries to Indexing and Search Service Web Services

The **isshttpd** proxy must be capable of mapping the user requesting the Indexing and Search Service search to its mail host. To do so, you can either create a flat file with the appropriate mappings, or add a new distinguished name to your Directory Server to perform the mapping.

If you choose to use a flat file, follow these guidelines:

- The file can reside on any file system in your deployment, including an NFS share.
- The file must be readable by the **iss.user**, configured when you run the **setup** script.
- The flat file format is as follows:

```
mailhost1.domain.com=isshost1.domain.com:isswebport
mailhost2.domain.com=isshost2.domain.com:isswebport
...
```

If you choose to use Directory Server, then when you run the Indexing and Search Service **setup** script, choose **ldap** for the lookup source. Choosing **ldap** creates a distinguished name called **Indexing and Search Service** in the Directory Server **comms-config** base object.

Installing and Configuring isshttpd

Installing and configuring **isshttpd** consists of the following high-level steps:

1. Deciding how to map user entries to look up Indexing and Search Service web services, by using either a flat file or Directory Server
2. Running the Indexing and Search Service **setup -t isshttpd** command
3. Configuring Indexing and Search Service to use the appropriate lookup method
4. Configuring Convergence to use **isshttpd**

Checking for o=comms-config in Directory Server

If you decide to use Directory Server to perform the Indexing and Search Service mapping, check that the **o=comms-config** base object is present in LDAP. This base object should already be present in a Directory Server deployment that uses

Messaging Server. The following example **ldapsearch** command searches the **ds.domain.com** domain for the **o=comms-config** base object:

```
ldapsearch -h ds.domain.com -p 389 -j /tmp/password -D "cn=Directory manager" -b
"o=comms-config" ""

version: 1
dn: o=comms-config
objectClass: top
objectClass: organization
o: comms-config
```

If the o**=comms-config** base object is not present in LDAP, you must add it.

After you have run the Indexing and Search Service **setup** script, you administer this LDAP entry either with the **isshttpdmgr** command, or through the **ldapmodify** command. Use the distinguished name "**cn=ISS, o=comms-config**".

setup Script Overview

The Indexing and Search Service **setup** script generates configuration files required to run the indexing services. The script uses the *IndexSearch_home***/etc/jiss.conf.template** file for default values but prompts for answers to various configuration questions about the Messaging Server messaging store installation and the local system services. For more information, see "setup Script."

You can deploy **isshttpd** as part of an existing Indexing and Search Service installation or in an independent environment. Depending on the installation type, the **setup -t isshttpd** command prompts you with different questions.

Running the setup Script

Before running the **setup** script to configure **isshttpd**, you might need to perform the following actions:

- If **isshttpd** is using a value of **file** for the **iss.isshttpd.lookup.source** parameter, create the file itself before running **setup**. Otherwise, the service does not start correctly. See "How isshttpd Maps User Entries to Indexing and Search Service Web Services" for the format to use when creating the file.
- Running the **setup** script creates the **iss.user** and **iss.group** items by using the next available UID and GID, if they are not present on the host. To ensure that the UID and GID of this user and group are consistent across deployments, create this user and group before running **setup**. For more information about **iss.user** and **iss.group**, see the topic on Indexing and Search Service configuration parameters in *Indexing and Search Service System Administrator's Guide*.

The following **setup -t isshttpd** examples configure **isshttpd** in an independent environment with no other Indexing and Search Service services running on the host. To configure other Indexing and Search Service services, you would run **setup** without the **-t isshttpd** option.

To run the **setup** command to configure **isshttpd** for file lookup:

```
IndexSearch_home/bin/setup -t isshttpd

(iss.cluster.install) [standalone]: <Accept the default of standalone>

iss.isshttpd.lookup.source [ldap]: <Choose file>

(iss.isshttpd.lookup.file) [/etc/jiss/isshttpd.lookup]: <Enter the location of
```

```
your lookup file>

(iss.isshttpd.bind.localhost) [true]: <isshttpd will only listen on localhost,
select if installing on the same host as Convergence>

(hostname) [none]: <Fully qualified domain name of this system>

(iss.user) [jiss]: <User under which all ISS services run>

(iss.group) [jiss]: <Group under which all ISS services run>

(iss.log.dir) [/var/opt/sun/comms/jiss/logs]: <Location of ISS log files>

(java.home) [/usr/jdk/latest]: <Location of Java>

Setup Isshttpd service
Performing a check of the ISS installation
All configurations have been run successfully
```

To run the **setup** command to configure **isshttpd** for LDAP lookup:

```
IndexSearch_home/bin/setup -t isshttpd

(iss.cluster.install) [standalone]: <Accept the default of standalone>

iss.isshttpd.lookup.source [ldap]: <Choose ldap>

(mail.ldap) [none]: <User/Group Directory Server>

(mail.defaultdomain) [none]: <User/Group Default Domain>

(mail.searchbind) [cn=Directory manager]: <User/Group Directory Manager DN>

(mail.searchbind.password) [none]: <User/Group Directory Manager password>

Setup Isshttpd service
Performing a check of the ISS installation
All configurations have been run successfully
```

Configuring Lookup Entries

This section describes how to configure lookup entries for the **isshttpd** proxy.

Configuring File-Based Lookup

To configure file-based lookup:

1. Add entries to a flat file location specified by the **iss.isshttpd.lookup.file** parameter.

   This file must be readable by the **iss.user** and **iss.group**, and include the following entries:

   ```
   mailhost1.domain.com=isshost1.domain.com:isswebport
   mailhost2.domain.com=isshost2.domain.com:isswebport
   ...
   ```

2. Create this file for each instance of **isshttpd**, unless you have located the file on an NFS share.

3. Restart the **isshttpd** service.

- Solaris:

  ```
  svcadm restart jiss-isshttpd
  ```

- Linux:

  ```
  /etc/init.d/isshttpd stop
  /etc/init.d/isshttpd start
  ```

Configuring LDAP-Based Lookup

When adding entries to the Directory Server under the distinguished name **cn=ISS, o=comms-config**, use the **isshttpdmgr** command.

To configure LDAP-based lookup:

1. Create a file that defines the mapping.

   Be sure to use the format *host,port*, for example:

   ```
   sunkeyvalue: mailhost1.example.com=isshost1.example.com,8080
   sunkeyvalue: mailhost2.example.com=isshost2.example.com,8080
   ```

2. Run the following command to add the entries into Directory Server.

   ```
   IndexSearch_home/isshttpd/scripts/isshttpdmgr -u file_name
   ```

3. You can list the current entries by using the following command:

   ```
   IndexSearch_home/isshttpd/scripts/isshttpdmgr -l
   ```

Configuring Convergence to Use isshttpd

To configure Convergence to use the **isshttpd** proxy and enable Indexing and Search Service:

1. Run the following **iwcadmin** commands:

   ```
   Convergence_base/sbin/iwcadmin -o ISS.host -v localhost
   Convergence_base/sbin/iwcadmin -o ISS.port -v 5559
   Convergence_base/sbin/iwcadmin -o ISS.proxyadminid -v mail.imap.admin.username
   Convergence_base/sbin/iwcadmin -o ISS.proxyadminpwd -v
   mail.imap.admin.password
   Convergence_base/sbin/iwcadmin -o ISS.requesttimeout -v 180
   Convergence_base/sbin/iwcadmin -o ISS.enable -v true
   ```

2. Restart the GlassFish Server domain in which Convergence is running.

Troubleshooting isshttpd

This section describes how to troubleshoot your **isshttpd** installation.

isshttpd Service Start Fails Until LDAP Entries Are Created

When configuring the first Indexing and Search Service node to use LDAP to store the Messaging Server-to-Indexing-and-Search-Service host mapping, the **isshttpd** service does not start until you run the **isshttpmgr.sh -u** *file* command. However, you cannot run this command before the **setup -t isshttpd** command because it requires configuration information provided in that command. Once the LDAP entries are created and replicated, **isshttpd** should start on subsequent nodes.

Troubleshooting isshttpd Proxy Using wget

You can create **wget** scripts to simulate Convergence behavior to Indexing and Search Service.

To troubleshoot **isshttpd** by using **wget**:

1. Edit the **/etc/wgetrc** file to point to **isshttpd** with the following entries:

   ```
   http_proxy = localhost:5559
   ftp_proxy = localhost:5559
   ```

2. Run the following shell script while updating the password for **indexeradmin**, the user name, and the mail host:

   ```
   #!/bin/sh
   [ -f /tmp/isshttpd.out ] && rm -f /tmp/isshttpd.out
   [ -f /tmp/cookie.out ] && rm -f /tmp/cookie.out
   host=localhost
   mailhost=ms1.example.com
   username=username
   password=password
   /usr/sfw/bin/wget \--save-cookies=/tmp/cookie.out \
                            \--keep-session-cookies \
                            \--server-response \
                            \--no-check-certificate \
                            \-O /tmp/isshttpd.out \
                            \--max-redirect 0 \
                            \--post-data="j_username=indexeradmin%3B$username&j_
   password=$password" \
                            \-t 1 \
                            "http://$host:8080/rest/j_security_check"

   /usr/sfw/bin/wget \-O /tmp/isshttpd.out \
                            \--load-cookies=/tmp/cookie.out \
                            \--no-check-certificate \
                            \--server-response \
                            \-t 1 \

   "http://$host:8080/rest/search?q=%2busername:$username%20%2bhostname:$mailhost
   %20%2bfolder:INBOX&c=100"
   ```

This shell script searches the inbox of the user provided through **isshttpd**. You can validate different mappings by using a different user and mail host combination.

8

Configuring Indexing and Search Service for High Availability

This chapter describes how to plan and deploy a highly available Oracle Communications Indexing and Search Service system.

Indexing and Search Service High Availability Overview

This section provides an overview of Indexing and Search Service high availability (HA), and the benefits and limitations.

About Indexing and Search Service High Availability

Indexing and Search Service provides the ability to make its search component highly available with the Cluster Search Service and a highly available Network File System (NFS), upon which you locate Indexing and Search Service indexes. When Indexing and Search Service search is unavailable from an Indexing and Search Service web node, the clients' search requests are redirected to another Indexing and Search Service web node that accesses the HA NFS and locates the appropriate index. Thus, the Indexing and Search Service search component can fail without an effective loss of the overall search functionality. Additionally, by using hardware load balancers in front of the Indexing and Search Service web nodes, you split the network load across these Indexing and Search Service front ends, increasing their availability to respond to client requests.

The Cluster Search Service can:

- Perform searches on known indexing repositories
- Manage account state information (removes dependency on topic **AccountState.***hostname* being responsive on indexing host)
- Manage user directory lookup information (removes dependency on queue **Index**.*hostname* being responsive on indexing host)

Availability is often shown as the percentage of time that the system is up, or available, by using a system of "nines." For example:

- 2 x 9's = 99% uptime = 89.2 hours downtime/year
- 2.5 x9's = 99.5% uptime = 44.6 hours downtime/year
- 3 x 9's = 99.9% uptime = 8.92 hours downtime/year
- and so on

Indexing and Search Service enables you to deploy a highly available indexing and search system with a goal of enabling at least a 99.9 percent uptime.

> **Note:** You can configure Indexing and Search Service for high availability such that the NFS tier, on which the indexes are stored, is not exposed through your firewall. In Indexing and Search Service terms, this means that you use the clusterv2 installation type to configure high availability. For more information, see "Configuring Indexing and Search Service for clusterv2".

Benefits and Limitations of Indexing and Search Service High Availability

The following summarizes the Indexing and Search Service HA benefits and limitations.

Benefits:

- Your organization can depend on the Indexing and Search Service software to perform its searches with little or no downtime.
- Reduced single points of failure.
- Ability to easily add more web and indexing hosts as your deployment grows.

Limitations:

- Does not provide protection from hardware or network failures.
- The performance of the NFS host (acts as the backup for indexes so that searches can continue when indexing hosts are unavailable) could impact searching performance.
- The longer indexing hosts are unavailable, the more out-of-sync the indexes become with the accounts.

Indexing and Search Service High Availability Model

The Indexing and Search Service High Availability Model is based on making the Indexing and Search Service search services themselves redundant. The Indexing and Search Service High Availability Model is not a hot-spare model where one node fails over to a standby node. The Indexing and Search Service High Availability Model is not a data replication model that enables any node to fail without loss of data.

Indexing and Search Service High Availability Architecture

The following components comprise an Indexing and Search Service architecture:

- Messaging Server: Message store, which contains the users' email and attachment data.
- Indexing and Search Service: Indexing service, which runs the bootstrapping operation and processes JMQ events from the message store.
- Indexing and Search Service: Search service, which runs the search services and GlassFish Server (web container) to process client search requests. The web container acts as the front end, or access layer, for search clients.

The Indexing and Search Service architecture also depends on a Directory Server for LDAP storage and access. The Directory Server contains user information for all message stores.

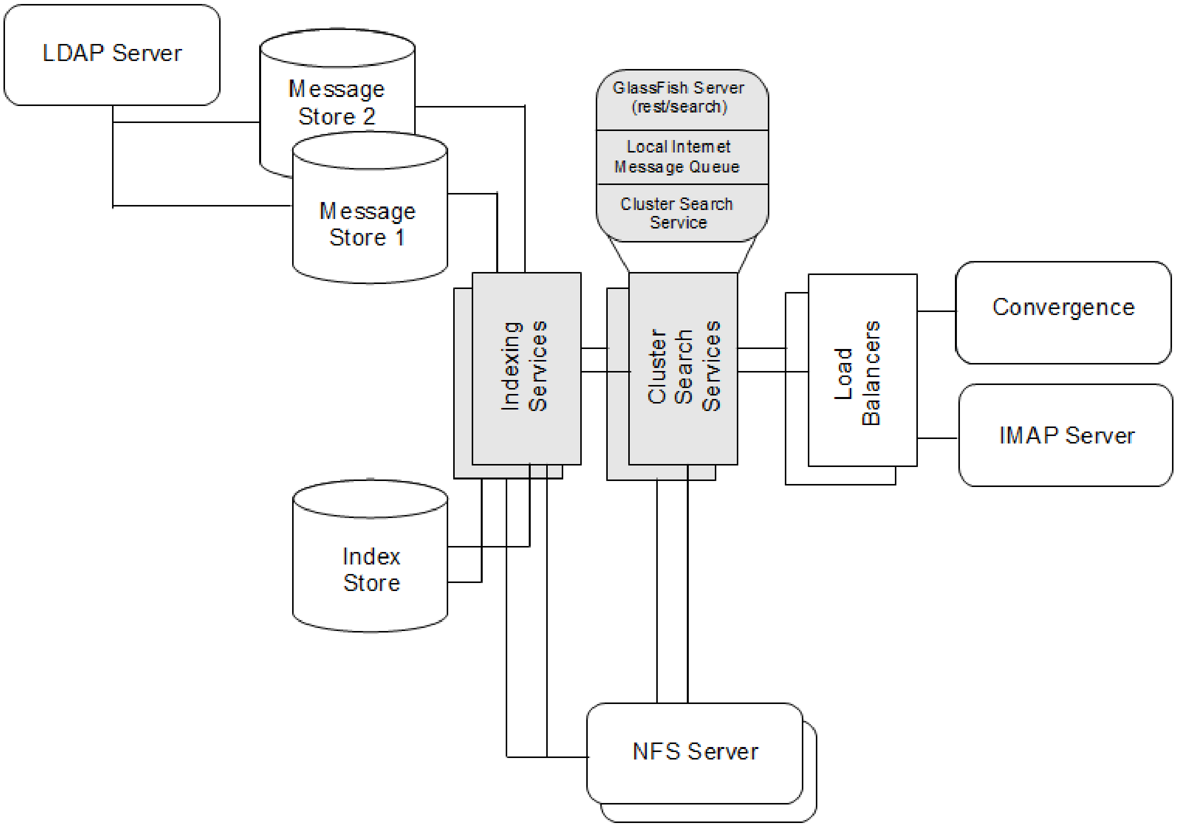
See Figure 2–1, "Indexing and Search Service Architecture" for the Indexing and Search Service logical architecture in non-HA mode.

To make the Indexing and Search Service deployment highly available, you use the following setup:

- GlassFish Server: Use hardware load balancers to split the network load across the GlassFish Server front-end hosts.
- Search service: Use Cluster Search Service, which can search and maintain account state information for different dIndex repositories.
- Indexes: Place indexes on an HA NFS to be shared to the Clustered Search Service in case the indexing host is unavailable.

Figure 8–1 shows the Indexing and Search Service HA architecture.

Figure 8–1 Indexing and Search Service HA Architecture



Migrating From a non-HA Indexing and Search Service Deployment to an HA Indexing and Search Service Deployment

To migrate a standalone Indexing and Search Service node to an HA indexing node requires the following operations:

1. Stop all indexing services.

```
IndexSearch_home/bin/svc_control.sh stop
```

2. Remove the Indexing and Search Service WAR files from GlassFish Server.

```
IndexSearch_home/bin/setup -t web -u
```

3. Stop GlassFish Server.

   It is no longer be needed on this host for Indexing and Search Service operation.

4. Update the following parameters in the *IndexSearch_home*/**etc/jiss.conf** directory:

   ```
   iss.cluster.enabled = true
   iss.cluster.type = index
   ```

5. Copy the **iss.store.dir** and **iss.attach.dir** to the NFS shares.

6. Mount the NFS shares so that the locations in the *IndexSearch_home*/**etc/jiss.conf** file for **iss.store.dir** and **iss.attach.dir** are accessible.

7. Start the indexing services.

   ```
   IndexSearch_home/bin/svc_control.sh start
   ```

8. Generate a cluster configuration for this indexing node.

   ```
   IndexSearch_home/bin/configure_etc.pl -C output_file
   ```

9. Copy the cluster file to each web node and enable by using the *IndexSearch_home*/**bin/csearchmgr.sh** command:

   ```
   csearchmgr.sh -A
   ```

Configuring Indexing and Search Service High Availability

This section contains the generic procedures to configure Indexing and Search Service HA. See "Indexing and Search Service Highly Available Example" for a sample that walks you through configuring the different components of an Indexing and Search Service HA deployment on Oracle Solaris.

Before You Begin

Ensure that you have performed the following steps before beginning the Indexing and Search Service HA configuration:

1. Install and configure your Messaging Server deployment (most sites already have deployed Messaging Server).

2. Set up your HA NFS. Many options exist for this requirement. Choose an NFS that best suits your site's requirements. Setup a file system for the index store and another for the attachment thumbnail store.

3. Install and configure your Indexing and Search Service deployment. Note the following additional package requirements:

   - Indexing and Search Service Web Host: The web host needs GlassFish Server and Message Queue (JMQ).
   - Indexing and Search Service Indexing Host: The indexing host needs JMQ, and Directory Server (LDAP), if you do not already have a Directory Server set up to perform JNDI lookups.

Note: When installing Indexing and Search Service through the Installer, you can also install JMQ on each type of Indexing and Search Service node. You must install GlassFish Server and Directory Server separately.

Setting Up NFS to Contain the Indexing and Search Service Indexes

Note: In case of indexing host failure, putting the indexes on NFS still provides the ability to do searches. However, the longer the indexing host is unavailable, the more risk you run of the indexes getting out of date with the store.

To set up NFS to contain the Indexing and Search Service indexes:

1. On the NFS server, and Indexing and Search Service web and index hosts, create **iss.user** and **iss.group** with the same UID and GID, and edit the **/etc/default/nfs** file to ensure that the **NFSMAPID_DOMAIN** value is accurate.

2. On all NFS clients (the Indexing and Search Service web and index hosts), edit the **/etc/default/nfs** file and ensure that the **NFSMAPID_DOMAIN** value is the value used on the NFS server host.

   By default, the **nfsmapid** uses the DNS domain of the system. This setting overrides the default. This domain is used to identify user and group attribute strings in the NFS protocol. Clients and servers must match with this domain for operation to proceed normally.

3. On the NFS server, enable NFS, create mount points for the Indexing and Search Service hosts, create ZFS pools for each Indexing and Search Service index host, create mount points for the Indexing and Search Service hosts, change the ownership and permissions of these mount points, and share the NFS file systems.

   a. Use the **svcadm** command to enable NFS. There must be an entry in the **/etc/dfs/sharetab** file for the NFS server daemon to start.

   b. Create a ZFS pool for each index host.

   c. Create ZFS file systems, two for each index host, one for the index store, and one for the attachment thumbnail store, to be used as mount points on the NFS server.

   d. Change ownership of these file systems to **iss.user**.

   e. Change permission on these file systems to **755**.

   f. Set the mount points for the ZFS file systems on the Indexing and Search Service index hosts.

   g. Share the file systems as NFS file systems so that the NFS clients can mount and access them.

      For security reasons, only the Indexing and Search Service index hosts need read/write access. The Indexing and Search Service web hosts, which would access the file systems for searching purposes, are given read-only access. The web hosts are also assigned anonymous access and **iss.user** UID access.

      Anonymous access and **iss.user** UID access might not be needed, depending on the user that runs the GlassFish Server. Running the GlassFish Server as the

iss.user and **iss.group** eliminates the need for anonymous UID access to the share.

4. Perform the following steps on the Indexing and Search Service web hosts.

   a. Create directories on each web host to be used as mount points on the NFS server.

   b. Change ownership of these directories to the **iss.user**.

   c. Set the mount points as read-only for these directories on the NFS server.

5. On each Indexing and Search Service index and web host, create the Indexing and Search Service data directory mount point, change the user and group owner to **iss.user**:**iss.group**, and mount it from the NFS server with read/write privileges. Also, create a local file system directory for logs and change the user and group owner to **iss.user**:**iss.group**.

   For example:

   ```
   mkdir -p /var/opt/sun/comms/jiss
   chown jiss:jiss /var/opt/sun/comms/jiss
   mount -o rw nfs-host.domain:/index-host_mount_point_directory
   /var/opt/sun/comms/jiss
   mkdir -p /var/iss/logs chown jiss:jiss /var/iss/logs
   ```

Configuring the Indexing Hosts

Perform the following steps on each Indexing and Search Service indexing host:

1. Run the Indexing and Search Service **setup** script.

   ```
   cd IndexSearch_home/bin
   setup
   ```

2. Configure the cluster setup by responding to the following prompts.

   a. Enable cluster configuration (**iss.cluster.enabled**):

      Enter **true**.

   b. Type of cluster configuration web or index (**iss.cluster.type**):

      Enter **index**.

   c. Fully qualified domain name of this system (**hostname**):

      Enter the fully qualified name of this host, for example, **bco04.example.com**.

   d. Instance name of the installation for an indexing node (**instance.name**):

      Enter the unique instance name used to identify topics and queues for this host, for example, **bco04**. The instance name is not required to match the mount point name.

   e. Location to store the Lucene indexes (**iss.store.dir**):

      This parameter specifies the mount point on the NFS file system for indexes. Enter **/var/opt/sun/comms/jiss/index**.

   f. Location of attachment data (**iss.data.dir**):

      This parameter specifies the mount point on the NFS file system for attachment data. Enter **/var/opt/sun/comms/jiss/attach**.

   g. Location of Indexing and Search Service log files (**iss.log.dir**):

You should keep the logs on the local disk. Enter **/var/iss/logs**.

3. Configure the mail server setup by responding to the following prompt: Comma-delimited list of mail server IPs corresponding to mail.server (**mail.server.ip**).

   Enter the mail server IP address or addresses, for example, **10.0.2.0,10.0.2.1**.

   The mail server parameters for this indexing host point to the mail server that it indexes. In every configuration the User/Group Directory Server information should be identical. For every mail server in the cluster you must add it to the list of **mail.server.ip**.

4. Configure the Message Queue setup by responding to the following prompts.

   a. Indexing and Search Service JMQ broker hostname(s) list, that is, host:7676,host2:7677 (**imq.host**):

   b. Username for Indexing and Search Service JMQ broker (**iss.imq.user**):

   c. Password for Indexing and Search Service JMQ user (**iss.imq.password**):

   d. Password for administrative user on Indexing and Search Service JMQ broker (**iss.imq.admin.password**):

5. Configure the Directory Server setup for JNDI by responding to the following prompts.

   a. Indexing and Search Service Directory Server host list host:port,host2:port2 (**ldap.host**):

   b. Indexing and Search Service Directory Manager DN; format: cn=Directory Manager (**java.naming.security.principal**):

   c. Indexing and Search Service Directory Server password (**ldap.password**):

6. Configure the service setup by responding to the following prompt: Storeui access method, disk for single machine, http for multi-machine (**iss.storeui.access.method**)

   - Enter **disk**. (**http** is for a different type of install.)

7. Repeat the preceding setup steps for each indexing host.

Configuring the Web Hosts

Perform the following steps on each Indexing and Search Service web host:

1. Run the Indexing and Search Service **setup** script.

```
cd IndexSearch_home/bin
setup
```

2. Configure the cluster setup by responding to the following prompts.

   a. Enable cluster configuration (**iss.cluster.enabled**)

      Enter **true**.

   b. Type of cluster configuration web or index (**iss.cluster.type**)

      Enter **web**.

3. Configure the local install settings by responding to the following prompts.

   a. Fully qualified domain name of this system (**hostname**)

Enter the FQDN, for example, **bco01.example.com**.

b. Instance name of the installation for a web node (**instance.name**):

   Enter the unique instance name used to identify topics and queues for this host, for example, **bco01**.

c. Location to store the Lucene indexes (**iss.store.dir**):

   Accept the default (**/var/opt/sun/comms/jiss/index**).

d. Location of attachment data (**iss.data.dir**)

   Accept the default (**/var/opt/sun/comms/jiss/attach**).

e. Location of Indexing and Search Service log files (**iss.log.dir**):

   This must be on a local disk, for example, enter **/var/iss/logs**.

4. Configure the mail server setup by responding to the following prompt: Comma-delimited list of mail server IPs corresponding to mail.server (**mail.server.ip**).

   Enter the mail server IP address or addresses, for example, **10.0.2.0,10.0.2.1**.

   In every configuration the User/Group Directory Server information should be identical. For every mail server in the cluster you must add it to the list of **mail.server.ip**.

5. Configure the GlassFish Server settings by responding to the following prompts.

   a. Directory location of the Application Server (**appserv.dir**)

      The default is **/opt/glassfish3/glassfish**.

   b. Application Server web port (**appserv.web.port**)

      The default is **8080**.

   c. Application Server administrative port (**appserv.admin.port**)

      The default is **4848**.

   d. Application Server domain name for deployment (**appserv.domain**)

      The default is **domain1**.

   e. Application Server administrative user (**appserv.admin.user**)

   f. Application Server administrative password (**appserv.admin.password**)

6. Configure JMQ settings by responding to the following prompts.

   a. Indexing and Search Service JMQ broker hostname(s) list, that is host:7676,host2:7677 (**imq.host**)

      For example, enter **bco01.example.com:7676**.

   b. Username for Indexing and Search Service JMQ broker (**iss.imq.user**)

   c. Password for Indexing and Search Service JMQ user (**iss.imq.password**)

   d. Password for administrative user on Indexing and Search Service JMQ broker (**iss.imq.admin.password**)

7. Configure the service setup by responding to the following prompt: Storeui access method, disk for single machine, http for multi-machine (**iss.storeui.access.method**)

   - Enter **disk**. (**http** is for a different type of install.)

8. Repeat for each web host.

Generating and Importing Cluster Configuration Files

Each Indexing and Search Service index node that is part of the cluster needs to have a configuration file generated and copied to the Indexing and Search Service web nodes. This configuration file informs the web node where its files are, how to connect to get account state updates, and how to search it. This file enables the web nodes to perform a search.

1. On the first index host, use the **configure_etc.pl -C** command to generate the Indexing and Search Service *IndexSearch_home***/etc/jiss.conf** file but with the host name, for example, **bco04.conf**.

   ```
   cd /opt/sun/comms/jiss/bin
   configure_etc.pl -C index-host1.conf
   ```

2. Copy this *index-host1***.conf** file to the **/opt/sun/comms/jiss/etc/cluster.d** directory on the web hosts.

3. On the next index host, use the **configure_etc.pl -C** command to generate the Indexing and Search Service *IndexSearch_home***/etc/jiss.conf** file but with that host's name, for example, **bco29.conf**.

   ```
   cd /opt/sun/comms/jiss/bin
   configure_etc.pl -C index-host2.conf
   ```

4. Copy this *index-host2***.conf** file to the **/opt/sun/comms/jiss/etc/cluster.d** directory on the web hosts.

5. Perform the following steps on the web hosts:

   Update the **iss.store.dir** and **iss.attach.dir** parameters in the configuration files to point to the indexing node.

   - *index-host1*.conf

     ```
     instance.name = index-host1
     imq.host = index-host1.domain:7676
     iss.imq.user = jmquser
     iss.imq.password = password
     ldap.host = bco04.example.com:389
     java.naming.security.principal = cn=Directory Manager
     ldap.password = password
     java.naming.security.authentication = simple
     # These must be set manually:
     iss.store.dir =  /index-host1/index
     iss.attach.dir = /index-host1/attach
     ```

   - *index-host2*.conf

     ```
     instance.name = index-host2
     imq.host = index-host2.domain:7676
     iss.imq.user = jmquser
     iss.imq.password = password
     ldap.host = index-host2.domain:389
     java.naming.security.principal = cn=Directory Manager
     ldap.password = password
     java.naming.security.authentication = simple
     # These must be set manually:
     iss.store.dir =  /index-host2/index
     iss.attach.dir = /index-host2/attach
     ```

6. Set the owner and access permissions on the ***.conf** files.

   ```
   cd /opt/sun/comms/jiss/etc/cluster.d
   chown jiss:jiss *.conf
   chmod 600 *.conf
   ```

Starting Cluster Search Services

On the Indexing and Search Service web hosts that are running the Cluster Search Service, perform the following commands:

```
cd IndexSearch_home/jiss/bin
csearchmgr.sh -A
```

For more information, see the **csearchmgr.sh** usage documentation in *Indexing and Search Service System Administrator's Guide*.

Indexing Users on Indexing Hosts

Use the **issadmin.sh --bootstrap** command to bootstrap users on the message store hosts.

For more information, see the **issadmin.sh** usage documentation in *Indexing and Search Service System Administrator's Guide*.

Verifying Users on Web Hosts

To verify that users can run searches on the web hosts:

1. On each Indexing and Search Service web host, log in as the bootstrapped user.

   In this example, log in to **bco01** and **bco22**.

2. Perform a search by using the RESTful interface.

   The search defaults to the default mail host. Thus, you might need to change the hostname parameter in the URL if the user resides on a different mail host. For example, on **bco01.example.com** the search URL might resemble the following for user **c1**:

   ```
   http://bco01.example.com:8070/rest/search?q=%2busername:c1%20%2bhostname:bco65
   .example.com%20%2battachment-type:at*&contentFormat=attachmentOnly&thumbnail=s
   &c=100
   ```

   You would then change the username and hostname fields in the URL to the following for user **u1**:

   ```
   http://bco01.example.com:8070/rest/search?q=%2busername:u1%20%2bhostname:bco10
   8.example.com%20%2battachment-type:at*&contentFormat=attachmentOnly&thumbnail=
   s&c=100
   ```

Adding an Additional Web Host

To add web host to the deployment:

1. Repeat the installation for the new node.
2. Copy the index node configuration files to the new node.
3. Run **csearchmgr.sh -A**.
4. Add the new node to the load balancer.

Removing a Web Host

To remove a web host from the deployment

1. Remove the node from the load balancer.
2. Run **csearchmgr.sh -D**.

Indexing and Search Service Highly Available Example

This example shows how to configure an Indexing and Search Service HA deployment consisting of the following hosts:

- Two front-end web hosts
- Two index hosts
- Two message store hosts
- One NFS host

Assumptions for This Example

This example assumes that you have already deployed Messaging Server and Indexing and Search Service (in non-HA mode), and have a host that can serve as the NFS mount.

Note the following additional package requirements:

- Indexing and Search Service Web Host: The web host needs GlassFish Server and Message Queue (JMQ).
- Indexing and Search Service Indexing Host: The indexing host needs JMQ, and Directory Server (LDAP), if you don't already have a Directory Server set up to perform JNDI lookups.

> **Note:** When installing Indexing and Search Service through the Installer, You can install JMQ for each type of Indexing and Search Service node. You must install GlassFish Server and Directory Server separately.

Setting Up NFS to Contain the Indexing and Search Service Indexes

1. On the NFS host, and Indexing and Search Service web and index hosts, create **iss.user** and **iss.group** with the same UID and GID.
2. On all NFS client hosts (Indexing and Search Service index and web notes), edit the **/etc/default/nfs** file and ensure that the **NFSMAPID_DOMAIN** value is the same.

   ```
   NFSD_SERVERS=1024
   NFSMAPID_DOMAIN=example.com
   ```

3. Set up the ZFS pools on NFS server and configure the shares on each web and index host. In the following:
   - **bco04** and **bco29** are Indexing and Search Service index hosts.
   - **bco01** and **bco22** are Indexing and Search Service web hosts.
   - **nc-agile** is the NFS host.

- **nc-agile.example.com**

```
svcadm enable svc:/network/nfs/server:default
mkdir -p /bco04 /bco29
chown jiss:jiss /bco04 /bco29
chmod 755 /bco04 /bco29
zfs create pool/bco04
zfs create pool/bco29
zfs set mountpoint=/bco04 pool/bco04
zfs set mountpoint=/bco29 pool/bco29
share -F nfs -o
rw=bco04.example.com,ro=bco01.example.com,ro=bco22.example.com,anon=100
/bco04
share -F nfs -o
rw=bco29.example.com,ro=bco22.example.com,ro=bco01.example.com,anon=100
/bco29
```

- **bco01.example.com and bco22.example.com**

```
mkdir -p /bco04 /bco29
chown jiss:jiss /bco04 /bco29
mount -o ro nc-agile.example.com:/bco04 /bco04
mount -o ro nc-agile.example.com:/bco29 /bco29
```

- **bc04.example.com**

```
mkdir -p /var/opt/sun/comms/jiss
chown jiss:jiss /var/opt/sun/comms/jiss
mount -o rw nc-agile.example.com:/bco04 /var/opt/sun/comms/jiss
mkdir -p /var/iss/logs
chown jiss:jiss /var/iss/logs
```

- **bc029.example.com**

```
mkdir -p /var/opt/sun/comms/jiss
chown jiss:jiss /var/opt/sun/comms/jiss
mount -o rw nc-agile.example.com:/bco29 /var/opt/sun/comms/jiss
mkdir -p /var/iss/logs
chown jiss:jiss /var/iss/logs
```

Running the setup Script on Indexing Hosts

Perform the following steps on each Indexing and Search Service indexing host:

1. Run the Indexing and Search Service **setup** script.

```
cd IndexSearch_home/bin
setup
```

2. Configure the cluster setup.

```
Cluster setup
Enable cluster configuration (iss.cluster.enabled) [false]: true
Type of cluster configuration web or index (iss.cluster.type) [none]: index
Local Install Settings
Fully qualified domain name of this system (hostname) [none]:
bco04.example.com
Instance name of the installation for an indexing node, that is iss1.
(instance.name) [none]: bco04
Location to store the Lucene indexes (iss.store.dir)
[/var/opt/sun/comms/jiss/index]: /var/opt/sun/comms/jiss/index
Location of attachment data (iss.data.dir) [/var/opt/sun/comms/jiss/attach]:
/var/opt/sun/comms/jiss/attach
```

```
Location of JISS log files (iss.log.dir) [/var/opt/sun/comms/jiss/logs]:
/var/iss/logs
```

3. Configure the mail server setup.

   The mail server parameters for this indexing host point to the mail server that it indexes. In every configuration the User/Group Directory Server information should be identical. For every mail server in the cluster you must add it to the list of **mail.server.ip**.

   ```
   Comma-delimited list of mail server IPs corresponding to mail.server
   (mail.server.ip) [none]: 10.0.2.0,10.0.2.1
   ```

4. Configure the Message Queue setup.

   ```
   Message Queue Settings
   JISS JMQ broker hostname(s) list, that is, host:7676,host2:7677 (imq.host)
   [none]: bco04.example.com:7676
   Username for JISS JMQ broker (iss.imq.user) [jmquser]: jmquser
   Password for JISS JMQ user (iss.imq.password) [none]: changeme
   Password for admin user on JISS JMQ broker (iss.imq.admin.password) [none]:
   changeme
   ```

5. Configure the Directory Server setup.

   ```
   Directory Server Configuration for JNDI
   JISS Directory Server host list host:port,host2:port2 (ldap.host) [none]:
   bco04.example.com:389
   JISS Directory Manager DN; format: cn=Directory Manager
   (java.naming.security.principal) [cn=Directory Manager]: cn=Directory manager
   JISS Directory Server password (ldap.password) [none]: changeme
   ```

6. Configure the service setup.

   ```
   Service configuration
   Storeui access method, disk for single machine, http for multi-machine
   (iss.storeui.access.method) [disk]: disk
   ```

7. Repeat for each Indexing and Search Service indexing host.

Running the setup Script on Web Hosts

Perform the following steps on each Indexing and Search Service web host:

1. Run the Indexing and Search Service setup script.

   ```
   cd IndexSearch_home/bin
   setup
   ```

2. Configure the cluster setup.

   ```
   Cluster setup
   Enable cluster configuration (iss.cluster.enabled) [false]: true
   Type of cluster configuration web or index (iss.cluster.type) [none]: web
   ```

3. Configure the local install settings.

   ```
   Local Install Settings
   Fully qualified domain name of this system (hostname) [none]:
   bco01.example.com
   Instance name of the installation for an indexing node, ie iss1.
   (instance.name) [none]: bco01
   Location to store the Lucene indexes (iss.store.dir)
   ```

```
[/var/opt/sun/comms/jiss/index]: /var/opt/sun/comms/jiss/index
Location of attachment data (iss.data.dir) [/var/opt/sun/comms/jiss/attach]:
/var/opt/sun/comms/jiss/attach
Location of JISS log files (iss.log.dir) [/var/opt/sun/comms/jiss/logs]:
/var/iss/logs
```

4. Configure Mail Server settings.

   In every configuration the User/Group Directory Server information should be identical. For every mail server in the cluster you must add it to the list of **mail.server.ip**.

   In the Mail Server section mail server parameters every configuration for the User/Group Directory Server information should be identical. For every mail server in the cluster you must add it to the list of **mail.server.ip**.

```
Comma-delimited list of mail server IPs corresponding to mail.server
(mail.server.ip) [none]: 10.0.2.0,10.0.2.1
```

5. Configure GlassFish Server settings.

```
Application Server Settings
Directory location of the appserver (appserv.dir) [/opt/SUNWappserver]:
/opt/glassfish3/glassfish
Appserver web port (appserv.web.port) [8080]: 8080
Appserver admin port (appserv.admin.port) [4848]: 4848
Appserver domain name for deployment (appserv.domain) [domain1]: domain1
Appserver admin user (appserv.admin.user) [admin]: admin
Appserver admin password (appserv.admin.password) [none]: changeme
```

6. Configure JMQ settings.

```
Message Queue Settings
JISS JMQ broker hostname(s) list, that is host:7676,host2:7677 (imq.host)
[none]: bco01.example.com:7676
Username for JISS JMQ broker (iss.imq.user) [jmquser]: jmquser
Password for JISS JMQ user (iss.imq.password) [none]: changeme
Password for admin user on JISS JMQ broker (iss.imq.admin.password) [none]:
changeme
```

7. Configure the service setup.

```
Service configuration
Storeui access method, disk for single machine, http for multi-machine
(iss.storeui.access.method) [disk]: disk
```

8. Repeat for each Indexing and Search Service web host.

Generating and Importing Cluster Configuration Files

To generate and import the cluster configuration files:

1. On the first index host (this example uses **bco04.example.com**):

```
cd /opt/sun/comms/jiss/bin
configure_etc.pl -C bco04.conf
scp bco04.conf bco01:/opt/sun/comms/jiss/etc/cluster.d
scp bco04.conf bco22:/opt/sun/comms/jiss/etc/cluster.d
```

2. On the next index host (this example uses **bco29.example.com**):

```
cd /opt/sun/comms/jiss/bin
configure_etc.pl -C bco29.conf scp bco29.conf
```

```
bco01:/opt/sun/comms/jiss/etc/cluster.d scp bco29.conf
bco22:/opt/sun/comms/jiss/etc/cluster.d
```

3. On the web hosts (this example uses **bco01.example.com** and **bco22.example.com**):

   Update **iss.store.dir** and **iss.attach.dir** in configuration files.

   - **bco04.conf**

     ```
     instance.name = bco04
     imq.host = bco04.example.com:7676
     iss.imq.user = jmquser
     iss.imq.password = password
     ldap.host = bco04.example.com:389
     java.naming.security.principal = cn=Directory Manager
     ldap.password = password
     java.naming.security.authentication = simple
     # These must be set manually:
     iss.store.dir = /bco04/index
     iss.attach.dir = /bco04/attach
     ```

   - **bco29.conf**

     ```
     instance.name = bco29
     imq.host = bco29.example.com:7676
     iss.imq.user = jmquser
     iss.imq.password = password
     ldap.host = bco29.example.com:389
     java.naming.security.principal = cn=Directory Manager
     ldap.password = password
     java.naming.security.authentication = simple
     # These must be set manually:
     iss.store.dir = /bco29/index
     iss.attach.dir = /bco29/attach
     ```

4. Set the owner and access permissions on the ***.conf** files.

   ```
   cd /opt/sun/comms/jiss/etc/cluster.d
   chown jiss:jiss *.conf
   chmod 600 *.conf
   ```

Starting Cluster Search Services

On the web hosts that are running the Cluster Search Service, perform the following commands (in this example, **bco01.example.com** and **bco22.example.com**):

```
cd /opt/sun/comms/jiss/bin
csearchmgr.sh -A
```

Other useful **csearchmgr.sh** commands:

- List cluster search service entries:

  ```
  csearchmgr.sh -l
  ```

- Remove all cluster search services:

  ```
  csearchmgr.sh -D (remove all cluster search services)
  ```

- Add cluster search services for *name*:

  ```
  csearchmgr.sh -a -n name
  ```

- Delete cluster search services for *name*:

  ```
  csearchmgr.sh -d -n name
  ```

Indexing Users on Indexing Hosts

Run the following commands on the indexing hosts, to bootstrap users on the message store hosts.

For example, on **bco04.example.com**:

```
cd /opt/sun/comms/jiss/bin
issadmin.sh --bootstrap --host bco108.example.com --user u1
```

On **bco29.example.com**:

```
cd /opt/sun/comms/jiss/bin
issadmin.sh --bootstrap --host bco65.example.com --user c1
```

Verifying Users on Web Hosts

1. On each web host, log in as the bootstrapped user.

   In this example, log in to **bco01** and **bco22**.

2. Perform a search.

   The search defaults to the same mail host. Thus, you might need to change the hostname parameter in the URL if the user resides a different mail host. For example, on **bco01.example.com** the search url might resemble the following for user **c1**:

   ```
   http://bco01.example.com:8070/rest/search?q=%2busername:c1%20%2bhostname:bco65
   .example.com%20%2battachment-type:at*&contentFormat=attachmentOnly&thumbnail=s
   &c=100
   ```

   You would then change the username and hostname fields in the URL to the following for user **u1**:

   ```
   http://bco01.example.com:8070/rest/search?q=%2busername:u1%20%2bhostname:bco10
   8.example.com%20%2battachment-type:at*&contentFormat=attachmentOnly&thumbnail=
   s&c=100
   ```

Troubleshooting Indexing and Search Service High Availability

This section contains the following topics:

- General Differences Between Indexing Hosts and Web Hosts
- Troubleshooting Web Hosts
- Troubleshooting Indexing Hosts

General Differences Between Indexing Hosts and Web Hosts

Table 8–1 shows the two main differences, other than host name, in the **jiss.conf** file between the indexing hosts and web hosts. The installer should automatically configure these parameters based on the **iss.cluster.enable** parameter and type of node being configured.

Table 8–1 jiss.conf File Differences Between Indexing and Web Hosts

Parameter Name	Indexing Host	Web Host
java.naming.factory.initial	**com.sun.jndi.ldap.LdapCtxFactory**	**com.sun.jndi.fscontext.RefFSContextFactory**
iss.accountstate.dst.name	**AccountState.***instance.name*	**AccountState**

The naming factory change tells the web host to use file-based JNDI lookups that point to the local host's JMQ broker. The account state change is used to create a single point where all account state updates are funneled through a single topic to the GlassFish Server WAR files (**rest** and **storeui**).

Troubleshooting Web Hosts

To troubleshoot Indexing and Search Services web hosts:

1. Verify that the Cluster Search Services are running.

```
svcs -a | grep jiss
online         Jan_19   svc:/application/jiss-csearchSvc-bco04:default
online         Jan_19   svc:/application/jiss-csearchSvc-bco29:default
```

2. Check the Cluster Search Services log files:

```
iss.log.dir/iss-csearchsvc-bco04.log.0
iss.log.dir/iss-csearchsvc-bco29.log.0
```

3. Check JMQ connections.

   For each running Cluster Search Service, you should see the following information.

   - 1 consumer to **SearchTopic** topic
   - 1 consumer to **Index_instance.name_ queue**
   - 1 consumer per configuration in **iss.cluster.d** to **AccountState.***instance.name* on indexing host (shown in the following output)
   - GlassFish Server WAR files produce the following connections to JMQ running on the web host
   - 1 consumer per configuration in **iss.cluster.d** to **AccountState**
   - 512 or **iss.rest.proxypool.size** producers to **SearchTopic** if at least one RESTful search has been issued (otherwise this is 0)

```
# imqcmd list dst
--------------------------------------------------------------------------------------------------
Name          Type     State      Producers      Consumers                 Msgs
                                Total Wildcard  Total Wildcard    Count Remote UnAck Avg Size
--------------------------------------------------------------------------------------------------
AccountState Topic    RUNNING   0     0          2     0           0    0       0    0.0
Indexbco04   Queue    RUNNING   0     -          1     -           0    0       0    0.0
Indexbco29   Queue    RUNNING   0     -          1     -           0    0       0    0.0
SearchTopic  Topic    RUNNING   512   0          2     0           0    0       0    0.0
mq.sys.dmq   Queue    RUNNING   0     -          0     -           0    0       0    0.0
```

Troubleshooting Indexing Hosts

To troubleshoot Indexing and Search Service indexing hosts:

1. Check the log files.

   ```
   iss.log.dir/iss-indexsvc.log.0
   ```

2. Check JMQ connections on indexing node.

   You should see the following information:

   1 consumer for **AccountState.***instance.name* topic for each cluster search service started plus one for **jmqconsumer**.

```
# imqcmd list dst
-------------------------------------------------------------------------------------------------
Name          Type     State       Producers      Consumers                Msgs
                                Total Wildcard  Total Wildcard    Count Remote UnAck Avg Size
-------------------------------------------------------------------------------------------------
AccountState.bco04 Topic RUNNING 0    0         3     0           0     0       0    0.0
```

Uninstalling Indexing and Search Service High Availability

To uninstall Indexing and Search Service, run **setup -u** on each node.

9

Configuring Indexing and Search Service for clusterv2

This chapter describes how to configure Oracle Communication Indexing and Search Service for high availability such that the NFS tier, on which the indexes are stored, is not exposed through your firewall. In Indexing and Search Service terms, this means that you use the **clusterv2** installation type to configure high availability.

clusterv2 Overview

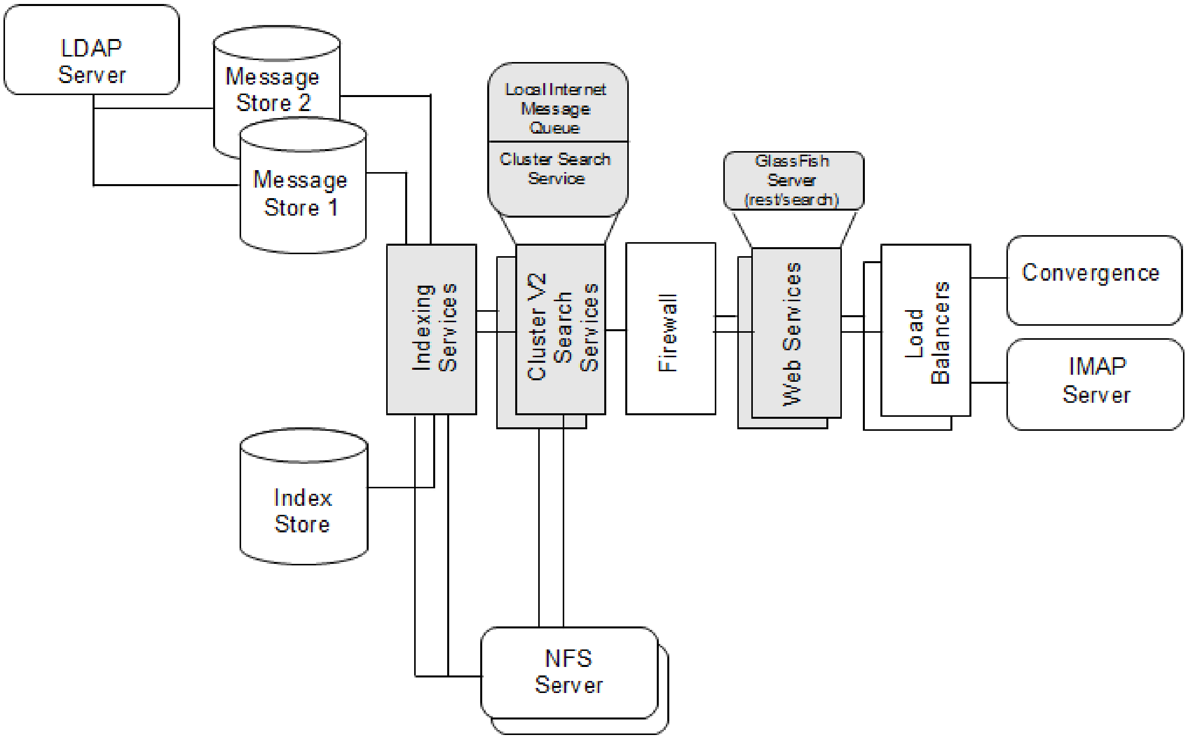
Indexing and Search Service provides the ability to make its search component highly available with the Cluster Search Service and a highly available Network File System (NFS), upon which you locate Indexing and Search Service indexes. When Indexing and Search Service search is unavailable from an Indexing and Search Service web node, the clients' search requests are redirected to another Indexing and Search Service web node that accesses the HA NFS and locates the appropriate index. Thus, the Indexing and Search Service search component can fail without an effective loss of the overall search functionality. Additionally, by using hardware load balancers in front of the Indexing and Search Service web nodes, you split the network load across these Indexing and Search Service front ends, increasing their availability to respond to client requests. This setup is described in "Configuring Indexing and Search Service for High Availability".

For sites interested in enhanced security, the **clusterv2** high availability installation type provides a solution where the NFS storage that contains the highly available indexes is protected behind your firewall. The front-end web nodes that handle the search requests are separated in front of the firewall from the searching and indexing nodes that are behind the firewall. Thus, requests to NFS are not allowed to pass through the firewall. This version of clustering uses HTTP.

clusterv2 Architecture

Figure 9–1 shows the Indexing and Search Service HA architecture using the **clusterv2** installation type.

Figure 9–1 Indexing and Search Service clusterv2 Architecture



In the preceding figure, the following setup is shown:

- GlassFish Server: Uses hardware load balancers to split the network load across the Web Services front ends (the **web** nodes).
- Firewall: Prevents NFS requests from coming through the front ends.
- Search service: Use Cluster Search Service (**clusterv2**), which can search and maintain account state information for different dIndex repositories.
- Indexes: Place indexes on an HA NFS to be shared to the Clustered Search Service in case the indexing host is unavailable.

Configuring clusterv2 High Availability

This section describes how to configure Indexing and Search Service HA by using the **clusterv2** installation type. You must install the NFS service and the **web**, **csearch**, and **index** Indexing and Search Service node types in the order shown.

Before You Begin

Ensure that you have performed the following steps before beginning the Indexing and Search Service HA **clusterv2** configuration:

1. Install and configure your Messaging Server deployment (most sites already have deployed Messaging Server).
2. Set up your HA NFS. Many options exist for this requirement. Choose an NFS that best suits your site's requirements.
3. Install and configure your Indexing and Search Service deployment. Note the following additional package requirements:

- Indexing and Search Service Web Host: The web host needs GlassFish Server.
- Indexing and Search Service Indexing Host: The indexing host needs Message Queue, and Directory Server (LDAP), if you don't already have a Directory Server set up to perform JNDI lookups.

> **Note:** When installing Indexing and Search Service through the Installer, you can install Message Queue for each type of Indexing and Search Service node. You must install GlassFish Server and Directory Server separately.

Node Types

Table 9–1 shows the different node types required for a highly available **clusterv2** configuration.

Table 9–1 clusterv2 Node Types

Node Name	clusterv2 value	Function	Required Service(s)
Web	**web**	Provides the Restful web interface	GlassFish Server
Cluster Search	**csearch**	Provides search layer (supports multiple index nodes)	Message Queue
Index	**index**	Provides indexing and real time event processing	Message Queue, Directory Server

Setting up NFS to Contain the Indexing and Search Service Indexes

> **Note:** In case of indexing host failure, putting the indexes on NFS still provides the ability to do searches. However, the longer the indexing host is unavailable, the more risk you run of the indexes getting out of date with the store.

To set up NFS to contain the Indexing and Search Service indexes:

1. On the NFS host, create two NFS shares, one for the index store and one for the thumbnail store. Make sure that **iss.user** and **iss.group** parameters have permissions to the files on the shares.
2. Mount the shares as read/write the on indexing host.

   These becomes the base directory for the **iss.store.dir** and **iss.attach.dir** parameters when configuring the index node.
3. Mount the shares as read-only on the cluster search host.

   These becomes the base directory for the **iss.store.dir** and **iss.attach.dir** parameters when configuring the **cluster.d** configuration file(s).

Configuring the Indexing Hosts

Perform the following steps on each Indexing and Search Service indexing host

1. If you haven't done so already, install Message Queue and Directory Server, which are the required services for an indexing host.

2. Run the Indexing and Search Service **setup** script.

   ```
   cd IndexSearch_home/bin
   setup
   ```

3. Configure the cluster setup by responding to the following prompts:
   - **iss.cluster.install**: **clusterv2**
   - **iss.cluster.type**: **index**
   - **iss.cluster.jndi.namespace**: *cluster_name* (Can be any alphanumeric text but must be the same for all nodes.)
   - **instance.name**: *issindex* (Can be any alphanumeric text but must be unique for all nodes.)
   - **iss.store.dir**: *nfsbase_directory*/**index**
   - **iss.attach.dir**: *nfsbase_directory*/**attach**
   - **iss.log.dir**: Place logs on the local file system.
   - **imq.host**: Message Queue server (usually running on the index node)
   - **jndi.type**: **ldap**
   - **ldap.host**: Directory Server (usually running on the index node)
   - **iss.storeui.access.method**: **disk**
4. Repeat the preceding setup steps for each indexing host.

Configuring the Search Hosts

To configure the Indexing and Search Service search hosts:

1. If you haven't done so already, install Message Queue, which is the required service for search hosts.
2. Run the Indexing and Search Service **setup** script.

   ```
   cd IndexSearch_home/bin
   setup
   ```

3. Configure the cluster setup by responding to the following prompts:
   - **iss.cluster.install**: **clusterv2**
   - **iss.cluster.type**: **csearch**
   - **iss.cluster.jndi.namespace**: *cluster_name* (Can be any alphanumeric text but must be the same for all nodes.)
   - **instance.name**: *isscsearch* (Can be any alphanumeric text but must be unique for all nodes.)
   - **iss.store.dir**: Accept default value, is not used.
   - **iss.attach.dir**: Accept default value, is not used.
   - **imq.host**: Message Queue Server (usually running on the cluster search node)
   - **iss.storeui.access.method**: **http**
4. Create the cluster configuration file for the index node, *IndexSearch_home*/**etc/cluster.d/iss1.conf**.

   a. On the index node run the following command:

```
IndexSearch_home/bin/configure_etc.pl -C /tmp/iss1.conf
```

   b. Copy the **iss1.conf** file to the cluster search node's *IndexSearch_home***/etc/cluster.d/iss1.conf** file.

   c. Update the **iss.store.dir** parameter to *nfsbase_directory***/index**.

   d. Update the **iss.attach.dir** parameter to *nfsbase_directory***/attach**.

5. Run the following command.

   ```
   IndexSearch_home/bin/csearchmgr.sh -A
   ```

6. Run the following command.

   ```
   IndexSearch_home/bin/configure_web_node -a WEB_NODE_IP_ADDRESS
   ```

Configuring the Web Hosts

To configure the Indexing and Search Service web hosts:

1. If you haven't done so already, install GlassFish Server, which is the required service for web hosts.

2. Configure the **clusterv2.conf** file:

   - Edit the *IndexSearch_home***/etc/cluster.d/clusterv2.conf** as follows:

     – **instance.name**: Comma-delimited list of index instance names, for example, **issindex**

     – **imq.host**: Comma-delimited list of cluster search Message Queue hosts, for example, **csearch.example.com:7676**

   - Change the ownership of the **clusterv2.conf** file:

     ```
     chown iss_user:iss_group IndexSearch_home/etc/cluster.d/clusterv2.conf
     ```

3. Run the Indexing and Search Service **setup** script.

   ```
   cd IndexSearch_home/bin
   setup
   ```

4. Configure the cluster setup by responding to the following prompts:

   - **iss.cluster.install**: **clusterv2**
   - **iss.cluster.type**: **web**
   - **iss.cluster.jndi.namespace**: *cluster_name* (Can be any alphanumeric text but must be the same for all nodes.)
   - **instance.name**: *issweb* (Can be any alphanumeric text but must be unique for all nodes)
   - **iss.store.dir**: Accept the default value, it is not used.
   - **iss.attach.dir**: Accept the default value, it is used.
   - **imq.host**: Message Queue server running on the cluster search node.
   - **iss.storeui.access.method**: **http**

Troubleshooting a clusterv2 Installation

This section contains common problems and their solutions when installing and configuring clusterv2 high availability.

"cannot findcn=CommsQueueFactory, cn=CommsTopicFactory, or cn=SearchTopic"

If the GlassFish Server (web node) contains the message "cannot findcn=CommsQueueFactory, cn=CommsTopicFactory, or cn=SearchTopic", there is a problem setting up the local JNDI information. Check the following settings:

- The **/etc/jiss/cluster.d/clusterv2.conf** file is present.
- Run *IndexSearch_home*/**bin/csearchmgr.sh -A**.
- Run *IndexSearch_home*/**bin/factorymgr.sh -l**, the output should be the following:

```
Listing all administered objects in the object store specified by:

 java.naming.factory.initial com.sun.jndi.fscontext.RefFSContextFactory
 java.naming.provider.url file:/etc/jiss/jms

 Lookup Name Object Class Name
 cn=CommsQueueFactory com.sun.messaging.QueueConnectionFactory
 cn=CommsTopicFactory com.sun.messaging.TopicConnectionFactory
 cn=IndexXXXX com.sun.messaging.Queue
 cn=SearchTopic com.sun.messaging.Topic

 Objects listed successfully.
```

Where *IndexXXXX* is the instance name from the **clusterv2.conf** file.

Searching Rest Interface Always Returns 404 Error

If you receive this error, check the following items:

- Ensure that the cluster search service is running on the cluster search node.

```
Java_home/bin/jps -l | grep com.sun.comms.iss.csearch.CSearchService
```

- View the GlassFish Server log on the web node while updating the state on the indexing node by running the **issadmin.sh --setstate A --user** command.
- View the cluster search log on the cluster search node while updating the state on the indexing node by running the **issadmin.sh --setstate A --user** command.

10

Indexing and Search Service Post-Installation Tasks

This chapter provides instructions for Oracle Communications Indexing and Search Service post-installation tasks.

Bootstrapping Users

After installing and configuring Indexing and Search Service, you must create, or "bootstrap," each user's account into the Indexing and Search Service store. You can bootstrap users either by manually running the **issadmin.sh --bootstrap** command or by enabling account autoprovisioning.

The following examples show how to manually run the **issadmin.sh --bootstrap** command.

To bootstrap a single user:

```
issadmin.sh --bootstrap --user username --host hostname --runoptimizer true
```

where:

- *username* specifies the user name of an account
- *hostname* specifies the host (or domain) name of an account

To bootstrap list of users, 16 users at time:

```
issadmin.sh --bootstrap --accountlist /tmp/accountlist.txt --threads 16
--runoptimizer true --altoutput /tmp/parallelbootstrap.log
```

where:

- *accountlist.txt* specifies a file containing account information for one or more accounts used to repeat the command action over multiple accounts

For more information about bootstrapping, see the topic on setting up and managing accounts in *Indexing and Search Service System Administrator's Guide*.

Configuring Messaging Server to Use Indexing and Search Service for IMAP SEARCH

Initially, when you prepared Messaging Server for Indexing and Search Service, you did not enable IMAP SEARCH queries to be redirected to Indexing and Search Service. After you bootstrap users and verify that the search works, you enable IMAP SEARCH queries to be redirected to Indexing and Search Service.

To configure Messaging Server to redirect IMAP SEARCH queries to Indexing and Search Service:

1. On the host where Messaging Server is installed, log in as or become the superuser (**root**).
2. Set the value for the **service.imap.indexer.enable** parameter to **1**:

   ```
   cd MessagingServer_home/bin
   configutil -o service.imap.indexer.enable -v 1
   ```

3. Refresh the Messaging Server configuration:

   ```
   refresh imap
   ```

If necessary, refer to "Preparing Messaging Server for Indexing and Search Service Integration" to prepare the Messaging Server host.

Verify Indexing and Search Service Services

The **check_conf.sh** script verifies that the necessary Indexing and Search Service services are running. The utility reads the *IndexSearch_home*/**etc/jiss.conf** file for configurations.

To run the **check_conf.sh** script:

1. On the host where Indexing and Search Service is installed, log in as or become the superuser (**root**).
2. Change to the *IndexSearch_home*/**bin** directory.
3. Run the **check_conf.sh** script.

   The following example shows that all Indexing and Search Service services are running.

   ```
   check_conf.sh
   No type defined (jmq, ldap, web) default to all
   Hostname match found
   test3.example.com:4848 (tcp) => Active
   test3.example.com:80 (tcp) => Active
   checking test3.example.com 389
   checking test3.example.com 7676
   Version string matched in headers: 101 imqbroker 4.[2-9]
   test3.example.com:143 (tcp) => Active
   checking test3.example.com 7676
   checking test3.example.com 389
   ```

For more information, see "check_conf.sh Script".

Verifying That Users Are Bootstrapped

To check that Indexing and Search Service is bootstrapping users:

1. On the Indexing and Search Service host, run the following command:

   ```
   IndexSearch_home/bin/issadmin.sh --listbrief > /tmp/listbrief.output
   ```

2. Look for the user names in the **/tmp/listbrief.output** file.

   If the account exists, but is not in state A (active), then it is not searchable by Indexing and Search Service and Messaging Server is performing the search.

Output from the **listbrief** command resembles the following:

```
Store ID: ISSID_67257f4a-0011-46ef-871f-7a859a004925 created: 20121020041929
default host name: iss01.example.com
default export/import path: <none specified>
total size of store instance: 0
total number of accounts: 120062
total number of account groups: 120062
last known consecutive group: 162
total search queries performed: 0
total search query failures: 0
total index events processed: 26746
last backup performed: Thu Oct 25 02:45:20 GMT 2012
Path: /var/iss/index/store/dIndex.backupA
last host number: 1
last user number: 120062
attachment store enabled: true
dIndex memory locked: false
State User UserNumber Host HostNumber State Changed Created BootStrap Time

A test@example.com u120001 iss01.example.com h1 20121020053630 20121020053622
8810
A admin u120002 iss01.example.com h1 20121020053635 20121020053634 1650
[...]
```

This example output indicates that users **admin** and **test@example.com** are in A (active) state and so are searchable by Indexing and Search Service.

Testing Search

To test that Indexing and Search Service is searching correctly:

1. Search a user's content by accessing it in a browser at the following URL:

   `http://`*`GlassFish_host`*`:`*`port`*`/rest`

   Log in as the user or proxy authenticate as the read-only store administrative user (use the values that you configured for the **mail.imap.admin.username** and **mail.imap.admin.password** parameters).

2. To view a demo web application utilizing the Indexing and Search Service API, try the SearchUI link or access the following URL in a browser:

   `http://`*`GlassFish_host`*`:`*`port`*`/searchui`

   Try a search such as **+body:java** to search for attachments from email with the word "java" in the body.

3. Send a new email message to an indexed user and confirm that the new message appears in the search.

You can also ensure that the search capability is working correctly by sending email with attachments to a user then running the **searchRun.sh** script, specifying that user name. For more information about the **searchRun.sh** script, see *Indexing and Search Service System Administrator's Guide*.

Configuring Convergence to Use Indexing and Search Service

To configure Convergence to use Indexing and Search Service:

1. On the host where Convergence is installed, log in as or become the superuser (**root**).
2. Change to the *Convergence_home*/**sbin** directory.
3. Run the following commands to configure Convergence to provide the Indexing and Search Service attachment interface:

   ```
   iwcadmin -w password -o ISS.enable -v true
   iwcadmin -w password -o ISS.enablessl -v true (if SSL is used)
   iwcadmin -w password -o ISS.host -v ISS_web_frontend_FQDN
   iwcadmin -w password -o ISS.port -v ISS_web_frontend_port
   iwcadmin -w password -o ISS.proxyadminid -v mail_imap_admin_username
   iwcadmin -w password -o ISS.proxyadminpwd -v mail_imap_admin_password
   ```

   where:

   - *password* is the password for the Convergence administrative user.
   - *ISS_web_frontend_FQDN* is the fully qualified domain name of the Indexing and Search Service front-end host.
   - *ISS_web_frontend_port* is the web component port number on which Indexing and Search Service is deployed (and should be the same as the port number for **appserver.web.port** in the **jiss.conf** file). Typically, the port number is 8080.
   - *mail_imap_admin_username* is the value of the Indexing and Search Service **mail.imap.admin.username** parameter. (The value of the Indexing and Search Service **mail.imap.admin.username** parameter in the **jiss.conf** file should be the same as the Message Sever **store.indexeradmins** value.)
   - *mail_imap_admin_password* is the value of the Indexing and Search Service **mail.imap.admin.password** parameter. (The value of the Indexing and Search Service **mail.imap.admin.password** parameter should be the same as the Message Server **indexeradmins** user password.)
4. If you are deploying Convergence and Indexing and Search Service on the same host and on the same GlassFish Server domain, you must increase the number of GlassFish Server request processing threads to two times the number of CPUs on the host.

   a. Determine the number of CPUs on the system.

      – Solaris:

      ```
      mpstat | grep -v CPU | wc -l
      ```

      – Linux:

      ```
      cat /proc/cpuinfo | grep processor | wc -l
      ```

   b. Set the number of request processing threads to two times the number of CPUs determined. If the number of CPUs is 4 or less, set this number to 8.

      – GlassFish Server 2:

      ```
      /opt/SUNWappserver/bin/asadmin set
      server.http-service.request-processing.thread-count = <2 x
      number-of-cpus>
      ```

      – GlassFish Server 3:

      ```
      /opt/glassfish3/bin/asadmin set
      server-config.network-config.transports.transport.tcp.acceptor-threads
      = (2 x number-of-cpus)
      ```

5. Restart GlassFish Server.

Testing Convergence

To test that Oracle Communications Convergence is working properly with Indexing and Search Service:

1. Log in to Convergence as one of your users.
2. Make sure that the **Messages** tab is selected.
3. Click **Attachments**.

   You should see a list of all attachment thumbnails in the user account.

Enabling Account Autoprovisioning

Autoprovisioning enables you to add new accounts to the Indexing and Search Service store automatically. By using autoprovisioning, you make account creation less prone to error than when manually bootstrapping accounts. However, unlike manually bootstrapping accounts with the **issadmin.sh** command, which gives you control over the group or singleton status for an account, autoprovisioning simply creates the account in the next available index group in the Indexing and Search Service store by using the default allocation policy.

To enable autoprovisioning:

1. Edit the **jiss.conf** file and set the **iss.autoprovision.enabled** configuration parameter to **`true`**.
2. Refresh the Indexing and Search Service configuration for the change to take place.

   ```
   issadmin.sh --refresh
   ```

For more information, see the topic on autoprovisioning accounts in *Indexing and Search Service System Administrator's Guide*.

Enabling Periodic Autosync

Periodic autosync checks every Indexing and Search Service account against Messaging Server accounts at regular intervals to detect differences, for example, from a reconstructed folder or lost event notifications, that Indexing and Search Service can then automatically correct. For more information, see the topic on enabling periodic autosync in *Indexing and Search Service System Administrator's Guide*.

To enable periodic autosync:

1. Edit the **jiss.conf** file and set the **iss.indexsvc.periodic.autosync.enabled** configuration parameter to **`true`**.
2. (Optional) Set the **iss.indexsvc.periodic.autosync.interval** configuration parameter to specify the number of seconds to wait from the start of one period until the start of the next autosync period.

   The default is 300 seconds.
3. (Optional) Set the **iss.indexsvc.periodic.autosync.count** configuration parameter to specify the number of accounts to process from the list of accounts yet to be synced during each period.

The default is 1000.

4. Refresh the Indexing and Search Service configuration for the change to take place.

   ```
   issadmin.sh --refresh
   ```

5. For tuning suggestions, see the topic on period autosync tuning controls in *Indexing and Search Service System Administrator's Guide*.

Enabling Automatic Bootstrapping of Missing Accounts

Accounts that have not been bootstrapped or autoprovisioned may cause event notifications of changes to such accounts to fail. To correct these accounts, you can use the autobootstrapping feature, which is useful if you want your deployment to bootstrap uninitialized accounts under such conditions. For more information, see the topic about enabling periodic autobootstrap in *Indexing and Search Service System Administrator's Guide*.

To enable periodic autobootstrap:

1. Edit the **jiss.conf** file and set the **iss.indexsvc.periodic.autobootstrap.enabled** configuration parameter to **`true`**.
2. (Optional) Set the iss**.indexsvc.periodic.autobootstrap.interval** configuration parameter to specify the number of seconds to wait from the start of one period until the start of the next autobootstrap period.

   The default is 300 seconds.
3. (Optional) Set the **iss.indexsvc.periodic.autobootstrap.count** configuration parameter to specify the number of accounts to process from the list of accounts yet to be bootstrapped during each period.

   The default is 500.
4. Refresh the Indexing and Search Service configuration for the change to take place.

   ```
   issadmin.sh --refresh
   ```

Enabling the Watcher Service

The Indexing and Search Service watcher service is designed for the implementation of highly available (HA) Indexing and Search Service nodes. However, the service itself is useful even when you do not use HA. The watcher service provides local host monitoring of Indexing and Search Service services and alerts you, with log file messages and email warnings, when it detects a service outage. Once alerted, you can take the appropriate corrective action, such as restarting a service.

To enable the watcher service:

1. On an Indexing and Search Service node, configure the appropriate **iss.cluster** and **iss.watcher** parameters in the **jiss.conf** file.
2. Configure the watcher service.

   ```
   IndexSearch_home/bin/watchermgr.sh -a -b /opt/sun/comms/jiss
   ```

See *Indexing and Search Service System Administrator's Guide* for a list of watcher parameters that you can configure.

11

Upgrading Indexing and Search Service

This chapter explains how to upgrade your existing system to the latest release of Oracle Communications Indexing and Search Service.

About Upgrading Indexing and Search Service

Upgrade your Indexing and Search Service back-end hosts first then upgrade your front-end hosts.

Upgrading to 1.0.5

Upgrading to Indexing and Search Service involves the following steps:

- Running the **commpkg upgrade** command
- Running the **postpatch_restart** command
- (Optional) Rebootstrapping accounts

Upgrading Indexing and Search Service (1.0.4 to 1.0.5)

To upgrade to 1.0.5, run the **commpkg upgrade** command on each host in your deployment. For more information, see "upgrade Verb Syntax".

When upgrading in silent mode, see also "Upgrading Shared Components in Silent Installation".

Running postpatch_restart

After upgrading to the new Indexing and Search Service version, run the **postpatch_restart** command. Because the **postpatch** process might be interactive, you must manually run **postpatch_restart**.

To run **postpatch_restart**:

```
cd IndexSearch_home/bin
postpatch_restart IndexSearch_home
```

For more information, see "postpatch_restart Script".

Upgrading Java

If you also upgraded from Java 6 to 7, you must rebootstrap accounts. Java 7 is recommended. If you generate Indexing and Search Service indexes while on Java 6, once you upgrade to Java 7, you should regenerate the indexes. For more information, see the topics on rebootstrapping accounts and migrating from Java 6 to Java 7 in *Indexing and Search Service System Administrator's Guide*.

Rebootstrapping Accounts After Upgrading

When upgrading from a version prior to Indexing and Search Service 1 Update 3, after the upgrade, you should rebootstrap all user accounts. This step is not necessary when upgrading from Indexing and Search Service 1 Update 3 or greater.

Here is one possible way to rebootstrap your accounts:

1. Use the Messaging Server **imquotacheck** command to get a list of all accounts on the system.

   ```
   cd /opt/sun/comms/messaging64/bin
   imquotacheck -a > quotafile
   ```

2. Delete all information in the redirected **quotafile** except the user IDs. Ensure that there is a single user ID on each line of the file.
3. Get a sense of the size of the users by running the following commands.
   - How many users?

     ```
     wc quotafile
     ```

   - Largest users by size, distribution of account sizes?

     ```
     imquotacheck | sort -k 3,3rn | less
     ```

   - Total mailbox size in Kbytes?

     ```
     mboxutil -l | awk '{ sum += $2} END {print sum}'
     ```

   You might want to save a copy of the **imquotacheck** and **mboxutil -l** output to Indexing and Search Service for later reference.
4. On Indexing and Search Service, remove the existing index:

   ```
   IndexSearch_home/examples/removeindex
   ```

5. Create the account list of users:

   ```
   IndexSearch_home/examples/createuserlist < quotafile
   ```

6. Re-initialize the index:

   ```
   IndexSearch_home/examples/initializeindex
   ```

7. Bootstrap all users in the **quotafile** in parallel:

   ```
   IndexSearch_home/examples/parallelbootstrap
   ```

   By default, the **parallelbootstrap** command bootstraps 16 users in parallel. Edit the file to change this setting.

12

Uninstalling Indexing and Search Service

This chapter describes how to uninstall Oracle Communications Indexing and Search Service.

Uninstalling Indexing and Search Service

The **commpkg uninstall** command enables you to uninstall Indexing and Search Service and shared components. However, the **commpkg uninstall** command does not remove OS patches or shared components installed by **commpkg install**.

To uninstall Indexing and Search Service:

1. Log in as **root**.
2. Change to the *InstallRoot*/**CommsInstaller/bin/** directory.
3. Run the **commpkg uninstall** command.
4. Choose Indexing and Search Service from the list of installed Communications Suite components.
5. When prompted, enter **Yes** to continue.
6. To remove the Indexing and Search Service generated indexing data:
   a. Run the **setup -u** command.
   b. Remove all Indexing and Search Service generated data by running the *IndexSearch_home*/**bin/scrub_index.sh -r** script.

      Removing the data could take a long time, depending on the size of the index.

13

Installing Patches

This chapter describes how to install patches on Oracle Communications Indexing and Search Service.

About Patching Indexing and Search Service

Indexing and Search Service patches are posted on the My Oracle Support website:

https://support.oracle.com

> **Important:** Always read the patch ReadMe file in its entirety before installing a patch.

Some patches contain fixes and functionality that may not be of any interest to you or may apply to features that you have not installed or purchased. Read the patch ReadMe file to determine if you must install the patch.

Some patches are password protected. To request the password to download a protected patch, open a Service Request on the My Oracle Support website.

Planning Your Patch Installation

Before installing a patch, verify your version of Indexing and Search Service and ensure that the patch is not already installed.

Oracle recommends scheduling your patch installation during non-peak hours to minimize the disruption to your operations.

Oracle recommends installing a patch on a test system with a copy of your production data before installing the patch on your production system. Test the patch by logging into Indexing and Search Service and verifying the version number of installed components.

Installing a Patch

Oracle Solaris 11 introduced the Image Packaging System (IPS) for software installs and updates. IPS changes the way patches are delivered, because IPS does not support the **patchadd** command. On Solaris 11 systems, you must use Automated Release Update (ARU) patches. These patches differ from the older SRV4 Sun-style patches, which are not supported on Solaris 11. You can use ARU patches on other Solaris releases as well. To install an ARU patch, you use the **commpkg upgrade** command.

Installing an ARU Patch

To install an ARU patch on Indexing and Search Service, run the following command:

```
commpkg upgrade
```

Installing an SRV4 Patch

To install an SRV4-style patch on Indexing and Search Service, run the following command:

```
patchadd
```

For more information, see the **patchadd** man page.

Installing a Linux Patch

To install a Linux patch on Indexing and Search Service, run the following command:

```
rpm -F rpmname
```

For more information, see the **rpm** man page.

A

Indexing and Search Service Scripts

This appendix provides information about the Oracle Communications Indexing and Search Service installation and configuration scripts.

Common Information

The Indexing and Search Service installation and configuration scripts are located in the *IndexSearch_home*/**bin** directory. To run these scripts, you must log in as or become the superuser (**root**).

setup Script

The **setup** script enables you to perform an initial configuration of your Indexing and Search Service deployment. The **setup** script configures and starts all services for Indexing and Search Service. This script expects all supporting services to be running, including:

- Oracle GlassFish Server
- Message Queue
- Oracle Communications Messaging Server
- Oracle Directory Server Enterprise Edition (for both Indexing and Search Service and Messaging Server)

Syntax and Examples

```
setup [-u] [-v] [-O] [-b basedir] [-c|-f conf] [-D]
      [-l logfile] [-t type]
      [-i isshost] [-L] [-S statefile] [-s]
      [ install|uninstall|reconfigure|restart] [-h]
```

Table A–1 describes the **setup** options.

Table A–1 setup Options

Option	Description
-u	Optional. Specifies to run in uninstall mode.
-v	Optional. Enables shell debugging.
-O	Optional. Overwrites the **jiss.conf file** and password configuration if present.
-b *basedir*	Required. Specifies the base directory of the installation.

Table A–1 (Cont.) setup Options

Option	Description
-c | **-f** *conf*	Optional. Specifies a file from which to read configuration information for the **jiss.conf** file.
-D	Dry run, checks the configuration files but does not configure. Output configuration to **-S** *statefile* is required.
-l *logfile*	Optional. Defaults to **/var/tmp/jiss-install.***pid*.
-t *type*	Optional. Defaults to **all**. Specifies the type of install or uninstall to perform, can be a comma separated list: ■ **all**: All components ■ **web**: Configure GlassFish Server ■ **jmq**: Configure Message Queue ■ **ms**: Configure Messaging Server ■ **ldap**: Configure Directory Server ■ **cluster**: Configure Cluster Services ■ **index**: Configure Index and start indexing services ■ **watcher**: Configure the watcher service
-i *isshost*	Optional. Fully qualified domain name to use for host name.
-L	Optional. Use **/bin/hostname** for the Indexing and Search Service machine name, **/bin/hostname** must return a fully qualified domain name.
-S *statefile*	Optional. Generates state file based on install.
-s	Optional. Installs but does not start index services.
install | **uninstall** | **reconfigure** | **restart**	Use **install** to install Indexing and Search Service. The **setup** script prompts for information if you do not provide a statefile. Use **uninstall** to uninstall Indexing and Search Service. Use **reconfigure** to uninstall the product, remove the existing configuration, and reinstall Use **restart** to stop and start Indexing and Search Service services, and call the **svc_control.sh** script.
-h	Optional. Prints help text.

To install and configure all services:

```
setup -t all
```

To uninstall services:

```
setup -u
```

To install services and take options from a configuration file:

```
setup -c my.conf
```

To install services and overwrite the existing configuration:

```
setup -O
```

To install services and use the host to answer Indexing and Search questions:

```
setup -i isshost.example.com
```

To set up only **ldap** and **jmq** configurations:

```
setup -t ldap,jmq
```

check_conf.sh Script

The **check_conf.sh** script verifies that the necessary Indexing and Search Service services are running. The script reads the *IndexSearch_home***/etc/jiss.conf** file for configuration information.

Syntax and Examples

```
check_conf.sh [-v] [-h] [-t type]
```

Table A–2 describes the **check_conf.sh** options.

Table A–2 check_conf.sh Options

Option	Description
-t *type*	Optional. Specifies the service type or component to check: ■ **all**: Checks all services and components. ■ **web**: Checks GlassFish Server configurations. ■ **jmq**: Checks Message Queue configurations. ■ **ldap**: Checks Directory Server configurations. ■ **index**: Checks Index and starts Indexing services. ■ **ms**: Checks message store configurations.
-v	Optional. Enables shell debugging.
-h	Optional. Prints help text.

Exit code 0 indicates a success. Exit code 1 indicates a failure. Options can appear in any order.

To verify that all Indexing and Search Service services are active:

```
check_conf.sh

No type defined (jmq, ldap, web) default to all
Hostname match found
test3.example.com:4848 (tcp) => Active
test3.example.com:80 (tcp) => Active
checking test3.example.com 389
checking test3.example.com 7676
Version string matched in headers: 101 imqbroker 4.[2-9]
test3.example.com:143 (tcp) => Active
checking test3.example.com 7676
checking test3.example.com 389
```

configure_etc.pl Script

The **configure_etc.pl** script generates the *IndexSearch_home***/etc/jiss.conf**, *IndexSearch_home***/etc/keystore.jks**, and *IndexSearch_home***/etc/stowg** files. This script reads the *IndexSearch_home***/etc/jiss.conf.template** file for default values. If no configuration file is provided at run time, you are prompted to enter all information. If a configuration file is provided at run time and required parameters are missing, you are prompted for a value. For more information on the Java KeyStore, see *Indexing and Search Service Security Guide*.

Syntax and Examples

```
configure_etc.pl [-b basedir] [-f conf_file] [-l]
                 [-i fqdn] [-O] [-A] [-t type] [-s] [-S file]
                 [-R file] [-C file]
```

Table A–3 describes the **configure_etc.pl** options.

Table A–3 configure_etc.pl Options

Option	Description
-b *basedir*	Required. Specifies the base directory of the installation.
-f *conf_file*	Optional. Specifies a file from which to read configuration information for the **jiss.conf** file.
-l	Optional. Use local host for **jmq** and **ldap** host names. **/bin/hostname** must return a fully qualified domain name.
-i *fqdn*	Optional. Fully qualified domain name to use for host name.
-O	Optional. Overwrites the current **jiss.conf**, **keystore.jks**, and **stowg** files.
-A	Optional. Prompts for all configurations.
-t *type*	Optional. Defaults to **all**. Specifies the type of install or uninstall to perform, can be a comma separated list: ■ **all**: All components ■ **web**: Configure GlassFish Server ■ **jmq**: Configure Message Queue ■ **ldap**: Configure Directory Server ■ **index**: Configure Index and start indexing services ■ **ms**: Configure Message Store ■ **cluster**: Configure a cluster
-S *statefile*	Saves information to state file.
-s	Optional. Skips validation steps.
-R *file*	Preserves all data and regenerates comments in file (requires **-f** *file* option).
-c *file*	Creates a cluster configuration file based on the **jiss.conf** and **keystore.jks** files.

To generate configuration:

```
configure_etc.pl
```

To generate configuration by overwriting current configuration:

```
configure_etc.pl -O
```

To read values from a configuration file:

```
configure_etc.pl -f my.conf
```

To generate new configuration file and prompt for all values:

```
configure_etc.pl -A -O
```

configure_web_node Script

The **configure_web_node** script updates the **httpd.conf** file running under the **iss.user** and **iss.group** context to allow or deny access to the Indexing and Search Service attachment store. Use this script only in multiple host deployments where the attachment store is not on the same host as GlassFish Server.

Syntax and Examples

```
configure_web_node [-a|-d] [IP|IP|IP ...]
```

Table A–4 describes the **configure_web_node** options.

Table A–4 configure_web_node Options

Option	Description
-a	Optional. Allows access for listed IP addresses.
-d	Optional. Removes access for listed IP addresses.
IP | *IP* | *IP* ...	The IP address(es) to allow or deny.

To add an IP address:

```
configure_web_node -a 10.10.10.10
```

To remove an IP address:

```
configure_web_node -d 10.10.10.10
```

postpatch_restart Script

The **postpatch_restart** script updates the system after an Indexing and Search Service patch has been applied. The script performs five basic operations:

- Stops all Indexing and Search Service services
- Merges the configuration file
- Updates the configuration with new parameters
- Starts all Indexing and Search Service services
- Deploys the Indexing and Search Service war files to GlassFish Server

Syntax and Examples

```
postpatch_restart [-h] [-v] [-s] [-S file] -f [file]
                  [-b basedir|basedir] -T timeout
```

Table A–5 describes the **postpatch_restart** options.

Table A–5 postpatch_restart Options

Option	Description
-b *basedir* | *basedir*	Sets the base directory of the Indexing and Search Service installation.
-S *file*	Generates state file for silent install based on this run. A state file is generated only if you are prompted for new information that has changed between releases.
-f *file*	Specifies the file to use that contains the configuration information for silent install.
-v	Uses verbose output.

Table A–5 (Cont.) postpatch_restart Options

Option	Description
-s	Patches but do not start indexing services. Disables services if already running.
-h	Prints help text.
-T *timeout*	Sets the *timeout* in seconds for **postpatch_restart** to wait before timing out. If set to **0**, *timeout* is unlimited.

To run **postpatch_restart**, create a state file, and set the base installation:

```
postpatch_restart -S /tmp/new.state /opt/sun/comms/jiss
```

verify_conf.pl Script

The **verify_conf.pl** script is used to verify that the *IndexSearch_home***/etc/jiss.conf** and *IndexSearch_home***/etc/jiss_passwd.conf** files contain required parameters and formatting.

Syntax and Examples

```
verify_conf.pl [-f jiss_config] [-v] [-h]
```

Table A–6 describes the **verify_conf.sh** options.

Table A–6 verify_conf.sh Options

Option	Description
-f *jiss_config*	Optional. Specifies the location of the **jiss.conf file**. The script assumes a location of**../etc/jiss.conf** if you do not provide one.
-v	Uses verbose output.
-h	Prints help text.

Exit code 0 indicates a success. Exit code 1 indicates a failure. Options can appear in any order.

B

commpkg Reference

This appendix provides information about the Oracle Communications Indexing and Search Service **commpkg** command.

Overview of the commpkg Command

The **commpkg**, command, also referred to as the Installer, comprises several commands (verbs) that enable you to install, uninstall, and upgrade Indexing and Search Service software and its shared components. The **commpkg** command is installed in the directory in which you extract the product software.

Syntax

```
commpkg [general_options] verb [verb_options]
```

Table B–1 describes the **commpkg** command general options.

Table B–1 commpkg Command General Options

Option	Description
-? or **--help**	Displays help.
-V or **--version**	Displays the Installer version.
--OSversionOverride	Overrides the operating-system version check.
--fixEntsys [**y** | **n**]	Fixes an invalid Sun Java Enterprise System (Java ES) **entsys symlink**, making the link point to the latest Java version upgraded by **commpkg**. The Java ES symlink is located in **/usr/jdk/entsys-j2se**. Choose **--fixEntsys y** to fix the Java ES symlink to the Java files. If you do not specify this switch, **commpkg** prompts you if the symlink is invalid. However, in silent mode, the default is not to fix the symlink (the equivalent of using a value of n). To fix the symlink in silent mode, type **commpkg install --fixEntsys y --silent** *INPUTFILE* on the command-line.

Table B–2 describes the **commpkg** command verbs.

Table B–2 commpkg Command Verbs

Verb	Description
install	Performs software installation.
uninstall	Uninstalls software but does not remove OS patches or shared components installed by **commpkg install**.

Table B–2 (Cont.) commpkg Command Verbs

Verb	Description
info	Displays product information on the paths (also known as *installroots*) where Indexing and Search Service is installed, and the products that are installed in those paths.
upgrade	Performs software upgrade.
verify	Verifies installed product.

install Verb Syntax

```
commpkg install [install options] [ALTROOT|name]
```

> **Tip:** Installing Only Shared Components: To install just the product's shared components, launch the Installer then prefix your product selection with a tilde (~). You can type multiple selections by using a comma to separate the entries.

Table B–3 describes the **commpkg install** verb options.

Table B–3 commpkg install Options

commpkg install Options	Description
-? or **--help**	Displays help.
-V or **--version**	Displays the Installer version.
--excludeOS	Does not apply operating system patches during product installation.
--excludeSC	Does not install, upgrade, or patch any shared components.
ALTROOT | *name*	Use this option to install multiple instances of the product on the same host or Oracle Solaris zone. You can also use this option to perform a side-by-side upgrade of the product. This option is available on Solaris only. Specifies an alternate root directory for a multi-instance installation. The *InstallRoot* (the top-level installation directory for all products and shared components) is the alternate root. If you specify a *name*, it will be a friendly name associated with the *ALTROOT* that is registered in the software list. If you specify the *name* and it exists in the software list, the corresponding *ALTROOT* is used. If you also specify the **--installroot** option, it must correspond to the entry in the software list. If you specify name and it does not exist in the software list, it is added to the software list. Specifying any *name* other than "" implies an **ALTROOT**. A value for *name* of "" is reserved for the default root.
--installroot	Specify location of INSTALLROOT, the top level installation directory for all products and shared components. The top-level installation directory for individual products are subdirectories under INSTALLROOT. Default is **/opt/sun/comms**.
--distro *path*	Specifies the *path* to packages or patches for the products. Default: Location of **commpkg** script

Table B–3 (Cont.) commpkg install Options

commpkg install Options	Description
--silent *INPUTFILE*	Runs a silent installation, taking the inputs from the *INPUTFILE* and the command-line arguments. The command-line arguments override entries in the *INPUTFILE*. Installation proceeds without interactive prompts. Use **--dry-run** to test a silent installation without actually installing the software. Specify **NONE** for *INPUTFILE* to run in silent mode without using an input file. When you specify **NONE**, the installation uses default values.
--dry-run or **-n**	Does not install software. Performs checks.
--upgradeSC [**y** | **n**]	Upgrades or does not upgrade shared components as required. If this option is not specified, you are prompted for each shared component that must be upgraded by using package removal and installation. Default: **n** **Caution:** Upgrading shared components by using package removal and installation is irreversible. However, if you do not upgrade required shared components, products might not work as designed. The **--excludeSC** flag has precedence over this flag.
--auditDistro	Audits the installation distribution to verify that the patches and packages matches the versions in the product files internal to the installer.
--pkgOverwrite	Overwrites the existing installation package. You might use this option when you are installing a shared component in a global zone where either the shared component does not exist in a global zone, or the shared component exists in the whole root zone. The default is not to override the existing package. In general, shared components should be managed in the global zone.
--components *comp1 comp2...*	A space delimited set of component products. Each product has mnemonic associated with it. Use **commpkg info --listPackages** to see the mnemonic for a product. In most shells you must escape the space between each mnemonic, that is, by adding double quotes around all the components.
--skipOSLevelCheck	(Solaris only) The Installer always checks that the operating system is at a certain minimum patch level. Using this option skips the check.

uninstall Verb Syntax

```
commpkg uninstall [verb_options] [ALTROOT|name]
```

Table B–4 describes the **commpkg uninstall** verb options.

> **Note:** A fast way to uninstall a product that was installed in an alternate root is to simply remove the entire alternate root directory.

Table B–4 commpkg uninstall Options

commpkg install Options	Description
-? or **--help**	Displays help.
-V or **--version**	Displays the Installer version.
--silent *INPUTFILE*	Runs a silent uninstall, taking the inputs from the *INPUTFILE* and the command-line arguments. The command-line arguments override entries in the *INPUTFILE*. Uninstall proceeds without interactive prompts. Use **--dry-run** to test a silent installation without actually installing the software.
--dry-run or **-n**	Does not install software. Performs checks.
ALTROOT | *name*	Use this option to uninstall multiple instances of the product on the same host or Oracle Solaris zone. You can also use this option to perform a side-by-side upgrade of the product. This option is available on Solaris only. Specifies an alternate root directory for a multi-instance uninstallation. The *InstallRoot* (the top-level installation directory for all products and shared components) is the alternate root. If you specify a *name*, it will be a friendly name associated with the *ALTROOT* that is registered in the software list. If you specify the *name* and it exists in the software list, the corresponding *ALTROOT* is used. If you also specify the **--installroot** option, it must correspond to the entry in the software list. If you specify name and it does not exist in the software list, it is added to the software list. Specifying any *name* other than “” implies an **ALTROOT**. A value for *name* of “” is reserved for the default root.

upgrade Verb Syntax

```
commpkg upgrade [verb_options] [ALTROOT|name]
```

Table B–5 describes the **commpkg upgrade** verb options.

Table B–5 commpkg upgrade Options

Options	Description
-? or **--help**	Displays help.
-V or **--version**	Displays the Installer version.
--excludeOS	Does not apply operating system patches during product upgrade.
--excludeSC	Does not install, upgrade, or patch any shared components.

Table B–5 (Cont.) commpkg upgrade Options

Options	Description
ALTROOT | *name*	This option is available on Solaris only. Specifies an alternate root directory during a multiple host installation. The *InstallRoot* (the top-level installation directory for all products and shared components) is the alternate root. If you specify a *name*, it is an "alias" associated with the alternate root that is registered in the software list. You can use this option to upgrade multiple product instances on the same host or Solaris zone. Additionally, you can use this option to perform a side-by-side product upgrade.
--distro *path*	Specifies the *path* to packages and patches for the products. Default path: Location of the **commpkg** command.
--silent *INPUTFILE*	Runs a silent upgrade, taking the inputs from the *INPUTFILE* and the command-line arguments. The command-line arguments override entries in the *INPUTFILE*. Upgrade proceeds without interactive prompts. Use **--dry-run** to test a silent upgrade without actually installing the software. Specify **NONE** for *INPUTFILE* to run in silent mode without using an input file. When you specify **NONE**, the upgrade uses default values.
--dry-run or **-n**	Does not upgrade software but performs checks. This option creates a silent upgrade file in the **/tmp** directory.
--upgradeSC [**y** | **n**]	Indicates whether to upgrade shared components as required. If this option is not specified, you are prompted for each shared component that must be upgraded by the package uninstall/install. Default: **n** **Caution:** Upgrading shared components is irreversible. However, if you do not upgrade required shared components, products might not work as designed. The **--excludeSC** flag has precedence over this flag.
--pkgOverwrite	This option is only for Solaris systems. Overwrites the existing installation package. You might use this option when you are installing a shared component in a global zone where either the shared component does not exist in a global zone, or the shared component exists in the whole root zone. The default is not to override the existing package. In general, shared components should be managed in the global zone.
--components *comp1 comp2...*	Specifies products to be upgraded. Separate each component product with a space. (Thus, the list is a space-delimited set.) To specify each component product, use the mnemonic associated with that product. To display a list of the mnemonics for all the component products, use the **commpkg info --listpackages** command.
--usePkgUpgrade	If the upgrade can be performed by using a patch or an in-place package upgrade, this option uses the in-place package upgrade. The default is to use a patch to upgrade, if possible. **Note:** Patches are used only on Solaris.

verify Verb Syntax

```
commpkg verify [verb_options] [ALTROOT|name]
```

> **Tip:** When verifying the package installation in an alternate root, be aware that Indexing and Search Service uses the operating system components installed in the default root. Some products might also use shared components in the default root. Thus, verify the package installation in the default root as well.

Table B–6 describes the **commpkg verify** verb options:

Table B–6 commpkg verify Options

Options	Description
-? or **--help**	Displays help.
-V or **--version**	Displays the Installer version.
--excludeOS	Do not verify operating system components.
--excludeSC	Do not verify shared components.
--components *comp1 comp2...*	A space delimited set of component products. Each product has mnemonic associated with it. Use **commpkg info --listPackages** to see the mnemonic for a product. In most shells you must escape the space between each mnemonic, that is, by adding double quotes around all the components.
ALTROOT | *name*	Use this option to verify multiple instances of the product on the same host or Solaris zone. This option is available on Solaris only. Specify *ALTROOT* or *name* for an alternate root directory on which to perform the package verification.

info Verb Syntax

```
commpkg info [verb_options] [ALTROOT|name]
```

Table B–7 describes the **commpkg info** verb options.

Table B–7 commpkg info Options

Options	Description
-? or **--help**	Displays help.
-V or **--version**	Displays the Installer version.
--clean	Removes entries in the software list. If *ALTROOT* | *name* is specified, the option removes the entry from the software list. If no *ALTROOT* | *name* is specified, the option removes all entries from the software list.
--listPackages	Lists the packages that comprise Indexing and Search Service, shared components, and operating system auxiliary products. This option also displays the mnemonic for Indexing and Search Service and components such as **comm_dssetup.pl**.
--verbose	Prints product information installed in the *installroots*. To print information for a specific *installroot*, run the following command: **commpkg info --verbose** *installroot*

Table B–7 (Cont.) commpkg info Options

Options	Description
--components *comp1 comp2...*	A space delimited set of component products. Each product has mnemonic associated with it. Use **commpkg info --listPackages** to see the mnemonic for a product. In most shells you must escape the space between each mnemonic, that is, by adding double quotes around all the components.

About the Alternate Root

You can install multiple copies of the same product version on the same Solaris machine or Solaris zone by using the alternate root feature of the **commpkg install** command. For example, you might deploy a host with an installation in the default root directory, **/opt/sun/comms**, and a second, separate installation in the **/opt/sun/comms2** alternate root directory. The alternate root installation directory is the top-level directory underneath which the Indexing and Search Service component product and shared components are installed in their respective directories.

Some possible uses for multiple installations include:

1. Performing a side-by-side upgrade.
2. Enabling an installation to be easily moved from one machine to another.

> **Note:** The alternate root feature is available only on Solaris. This feature is a "power user" feature. If you are interested in installing more than one instance of the same version of Indexing and Search Service on the same physical host, another option is to use Solaris zones. For more information, see "Installing Indexing and Search Service on Solaris Zones".

ALTROOT | name Syntax and Examples

You can use the optional *ALTROOT* | *name* option with the **commpkg install**, **commpkg upgrade**, **commpkg uninstall**, and **commpkg verify** commands. You use either *ALTROOT* or *name*. The *name* acts as an alias for the alternate root installation path. The *name* is registered in an internal software list maintained by the Installer. You can use *name* for the alternate root's path in commands that accept the alternate root. The distinction between the alternate root and name is that the alternate root always begins with a slash (/) whereas the name does not.

Syntax:

```
commpkg [install|upgrade|unistall|verify] [ALTROOT|name]
```

Example 1:

```
commpkg install /opt/sun/comms2
```

In this example, the path **/opt/sun/comms2** is the alternate root, which becomes the top-level directory underneath which Indexing and Search Service software and shared components are installed.

Example 2:

```
commpkg install Comms2
```

In this example, **Comms2** is the name for the alternate root. During the installation process, the Installer prompts you to type in the alternate root installation directory.

Example 3:

In this example, you avoid installing the shared components in the alternate root by using the **--excludeSC** option:

```
commpkg install --excludeSC /opt/sun/comms2
```

Example 4:

To install only the shared components, run the **commpkg install** command and select the product you want to install, but prepend a tilde (~).

For example, if you plan to install Indexing and Search Service in the alternate root, you select ~1 during the default installation. This tells the Installer to install the dependencies but not the product itself.

Notes on the *ALTROOT* | *name* command-line argument:

- Specifying a slash (/) as an alternate root is the same as specifying the default root installation directory. That is, specifying a slash is interpreted by the Installer as having specified no alternate root.
- Likewise, specifying "" as an alternate root is interpreted as having specified no alternate root. (The "friendly name" for the default alternate root is "".)
- If you specify the **--installroot** option in addition to *ALTROOT* | *name*, the two must match.
- Operating system patches are always installed into the default root (/). Some shared components are installed into the *ALTROOT* and some are installed into the default root (/).
- You can quickly uninstall an *ALTROOT* installation by using the **rm -r** command on the alternate root directory. The next time that you run the **commpkg info** command, the internal software list that maintains the alternate root information is updated.
- You can create as many alternate roots as you like. Running the **commpkg info** command reports on the various alternate roots.

Understanding the Difference Between ALTROOT and INSTALLROOT

The following concepts define an alternate root (ALTROOT):

- An alternate root directory is a Solaris feature that is used by the **commpkg** command to enable multiple product installations on the same host.
- The default alternate root is the standard root directory (/) and is always present.

The following concepts define an installation root (*InstallRoot*):

- An *InstallRoot* is the top-level umbrella installation path for Indexing and Search Service.
- On the default alternate root (that is, /), you can specify an *InstallRoot*.
- On an alternate root, the *InstallRoot* is the same as the alternate root.

Default Root

If you use the default root, the default *InstallRoot* is:

/opt/sun/comms/

Using Both Default Root and Alternate Root

Suppose you want to install one instance of Indexing and Search Service in the **/opt/sun/mycompany/comms/** directory, and another instance of the same product in the **/opt/sun/mycompany/comms2/** directory. You would use the following commands:

For the default root:

```
commpkg install --installroot /opt/sun/mycompany/comms
```

For the alternate root:

```
commpkg install /opt/sun/mycompany/comms2/
```

Running Multiple Installations of the Same Product on One Host: Conflicting Ports

By default, after you initially configure the product on alternate roots, the ports used by the different product installations are the same and thus conflict with each other.

This is not a problem if you install multiple installations of the same product on the same host but only intend to have one instance running at one time. For example, you may perform a side-by-side upgrade scenario in which you plan to stop the old instance before you start the new instance.

However, you may plan to test the new instance while the old instance is still running (and supporting end users). In this scenario, the ports are used simultaneously, and you must take steps to resolve the port conflicts.