Oracle® Communications Messaging Server

Installation and Configuration Guide

Release 8.0

July 2015

Oracle Communications Messaging Server Installation and Configuration Guide, Release 8.0

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

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Chapter 1. Configuration Worksheets - Messaging Server 8.0

Messaging Server 8.0 Worksheet

Print and fill out this worksheet to use when responding to the Messaging Server `configure` script configuration options.

Option	Default Value	Fill in Your Site's Value (to Respond to the Script)
Path for Data and Configuration Files	`/var/`*msg-svr-base*	
User Name for Server Processes	`mailsrv`	
Group Name for Server Processes	`mail` (if the User Name for Server Processes already exists, then the primary group for that User Name is used, and no option will be prompted for)	
Fully Qualified Host Name (FQHN)	*your host.your domain* For example: `myhost.west.sesta.com`	
Default mail domain name	*your domain*	
Hostname for LDAP Directory Server	blank (indicating the local hostname)	
LDAP administrator login	cn=Directory Manager	
LDAP administrator password	No default value	
Mail address for postmaster notices	`admin@`*your domain*	
Password for server administration	No default value	
Mail relay IP addresses (systems permitted to relay mail without authentication)	Not applicable	

Chapter 2. Installation Worksheets - Directory Server

Directory Server Settings Worksheet

Print and fill out this worksheet to use when responding to the Directory Server configuration options in the various product initial configurators.

Option	Default Value	Fill in Your Site's Value (to Respond to the Installer)
Instance Directory	`/var/opt/SUNWdsee/dsins1`	
Directory Instance Port	389	
Directory Instance SSL Port	636	
Directory Manager DN	`cn=Directory Manager`	
Directory Manager Password	NA	

Chapter 3. Directory Server Setup Script (comm_dssetup.pl)

Directory Server Setup Script (comm_dssetup.pl)

After you install a Communications Suite product and *before* you create initial configurations for these products, you must prepare Directory Server by using the Communications Suite Directory Setup Script, (`comm_dssetup.pl`).

Topics:

- Before You Run the `comm_dssetup.pl` Script
- Running the comm_dssetup.pl Script
- Manually Updating Schema Files
- Resolving Conflicting Calendar Server OID's in the LDAP Schema

Before You Run the `comm_dssetup.pl` Script

This section covers information you need to understand before running the `comm_dssetup.pl` script.

Topics in this section:

- What the `comm_dssetup.pl` Script Does
- Directory Server Considerations for the comm_dssetup.pl Script
- Information You Need to Gather Before you Run the comm_dssetup.pl Script
- About the comm_dssetup.pl choices for Directory Server root path name and instance
- About the comm_dssetup.pl Script Schema Choices
- Access Manager Considerations
- Attribute Indexes Created by the comm_dssetup.pl Script

What the `comm_dssetup.pl` Script Does

The `comm_dssetup.pl` script performs the following three steps:

1. Collects your choices for utility options.
   For a list of the specific information this step requests, see Information You Need to Gather Before you Run the comm_dssetup.pl Script.
2. Generates a shell script and LDIF file from your options choices that will be used to modify the LDAP directory.
   If you are not using an Oracle product for your directory server, or have customized your Directory Server, stop the process here without running the shell script. For further information, see Directory Server Considerations for the comm_dssetup.pl Script.
3. Runs the shell script created from your options choices. Your directory is modified accordingly.

At the end of each step, the utility asks you if you want to continue. No changes are made to the LDAP directory until the third step.

Directory Server Considerations for the comm_dssetup.pl Script

The following is a list of the considerations for your LDAP directory:

`comm_dssetup.pl` is a configuration tool that is for local LDAP instances servers. Thus,

- You must install the `comm_dssetup.pl` script on every machine on which a Directory Server resides.
- You must run the `comm_dssetup.pl` script on the same machine as your Directory Server. The tool runs locally for a specific instance (specified by path of directory server or path of instance).
- `comm_dssetup.pl` is installed into the "DirPrepTool-base", but can be run against any Directory Server instance on the local system. If you have multiple DIT's on one system, you can maintain and update one install of `comm_dssetup.pl`, and apply it to every Directory Server on the system.

`comm_dssetup.pl` must configure every Directory Server instance for the same DIT.

- A Directory Server must be installed, configured, and running before you run the `comm_dssetup.pl` script.
- If you add an additional machine that has Directory Server installed on it (such as a replica), at a future date, run the `comm_dssetup.pl` script against it, too.

If you have customized your LDAP directory, the following considerations might apply:

- If you have indexed some attributes, you might have to reindex those attributes after the `comm_dssetup.pl` script runs.
- If you have added other `.ldif` files (schema definitions), they should not be affected, so no action should be necessary. However, back up your custom schema definition files before running the `comm_dssetup.pl` script. As of `comm_dssetup.pl` 6.4p5, the old schema files are backed up to the `/var/tmp/dssetup_`*`timestamp/`*`save` directory.
- For all customizations, including the previous two, stop the `comm_dssetup.pl` script after it generates the script and before it actually updates the LDAP directory. Then inspect the script to evaluate how its proposed actions will affect your LDAP directory. Take whatever actions you think necessary to protect your customizations before running the script against your directory.

Information You Need to Gather Before you Run the comm_dssetup.pl Script

The `comm_dssetup.pl` script runs by first requesting information about your Directory Server. Prepare for this by gathering the information shown in the following table. To help you keep track of this information, use the `comm_dssetup.pl` Script Configuration Worksheet in *Unified Communications Suite Installation and Configuration Guide*.

Information Item Needed	Default Value
Directory Server root path name	The default depends on the Directory Server version detected. The `comm_dssetup.pl` script does attempt to heuristically determine the value.
Which instance of Directory Server to use? (If more than one.)	The default depends on the Directory Server version detected. The `comm_dssetup.pl` script does attempt to heuristically determine the value.
Directory Manager Distinguished Name (DN)	`"cn=Directory Manager"`
Directory Manager's Password	N/A
Directory Server being used for user/group data? (yes), or configuration data only? (no) Note that a configuration data only Directory Server is used only for Messaging Server 6.2 or earlier.	`yes`
User and group root suffix (if yes to previous question)	The default depends on what is detected. The `comm_dssetup.pl` script does attempt to heuristically determine the value.
Schema version? (pick one of the following) `1` - Schema 1 `1.5` - Schema 2 Compatibility Mode 2 - Schema 2 Native Mode For more information on how to choose a schema, see About the comm_dssetup.pl Script Schema Choices. If you have one version of the schema installed and want to upgrade to a higher level, refer to the *Sun Java System Communications Services 6 2005Q4 Schema Migration Guide* before running this utility.	2. However, if you run `comm_dssetup` again, it defaults to the value that you chose the previous time.
If you choose Schema 1 or 1.5, you will need a DC tree. If the DC tree does not yet exist, the `comm_dssetup.pl` script creates only the root suffix node, its does not create the rest of the DC tree. You must create the rest of your DC tree yourself.	`o=internet`. However, if you run `comm_dssetup` again, it defaults to the value that you chose the previous time.

About the comm_dssetup.pl choices for Directory Server root path name and instance

The combination of the Directory Server root path and the instance is used to create an absolute pathname to the Directory Server instance. For example, if your Directory Server instance resides under `/var/opt/sun/directory/slapd-varrius` then you should specify `/var/opt/sun/directory` for the Directory Server root path and `slapd-varrius` for the Directory Server instance. The reason for having two prompts to specify one absolute path is historical. Prior to Directory Server (DS) 6.x, DS had the concept of a "server root" under which all DS instances (as well as the DS binaries) reside. Starting with DS 6.x and later, the concept of the "server root" was removed. DS instances (as well as the DS binaries) do not all have to reside under a single umbrella "server root" directory.

About the comm_dssetup.pl Script Schema Choices

Communications Suite servers support the following schema choices:

- LDAP Schema 2 native mode
  Corresponds to `comm_dssetup.pl` script schema version choice 2. This is the default for a fresh installation.
- LDAP Schema 1
  Corresponds to the `comm_dssetup.pl` script schema version choice 1.
- LDAP Schema 2 compatibility mode
  Corresponds to `comm_dssetup.pl` script schema version choice 1.5.

If you are still trying to decide which schema to use, for further explanation, see *Unified Communications Suite Schema Reference* and *Sun Java System Communications Services 6 2005Q4 Schema Migration Guide*.

Access Manager Considerations

Starting with **Delegated Administrator 7**, Access Manager is no longer required for Schema 2.

Note
Do not use the Access Manager console to administer users. Use Delegated Administrator for administering Messaging Server, Calendar Server, and Contacts Server users.

Attribute Indexes Created by the comm_dssetup.pl Script

Attribute indexes improve the performance of search algorithms. The tool offers to index attributes. If you choose to do so, it will add indexes for all the Communications Suite products.

The following table lists all the attributes the `comm_dssetup.pl` script indexes, grouped by suffix category. It also lists the type of indexes created for each attribute. For more information about Directory Server indexing, see the Directory Server documentation.

Suffix	Attributes Indexed	Type of Indexes Added
User/Group	`mail`	`pres`, `eq`, `approx`, `sub`
	`mailAlternateAddress`	`pres`, `eq`, `approx`, `sub`
	`mailEquivalentAddress`	`pres`, `eq`, `approx`, `sub`
	`mailUserStatus`	`pres`, `eq`
	`member`	`eq`
	`ou`	`pres`
	`cosspecifier`	`pres`
	`groupid`	`pres`, `eq`, `sub`
	`icsCalendar`	`pres`, `eq`, `approx`, `sub`
	`icsCalendarOwned`	`pres`, `eq`, `approx`, `sub`
	`uniqueMember`	`eq`
	`memberOf`	`eq`, `sub`

	cn	eq
	mgrpUniqueId	eq
	deleted	pres, eq
	davuniqueid	pres,eq
	inetCos	eq
User/Group (additional for Schema 2)	inetDomainBaseDN	pres, eq
	sunPreferredDomain	pres, eq
	associatedDomain	pres, eq
	o	pres, eq
	mailDomainStatus	pres, eq
	sunOrganizationAlias	pres, eq
DC Tree (for Schema 1)	inetDomainBaseDN	pres, eq
	mailDomainStatus	pres, eq
	inetCanonicalDomainName	pres, eq
Personal Address Book (PAB) (o=pab) **Note: For old Address Book**	memberOfManagedGroup	pres, eq
	memberOfPAB	pres, eq
	memberOfPABGroup	pres,eq
	un	eq
New PAB (o=PiServerDb)	displayname	pres, eq, sub
	MemberOfPiBook	eq
	MemberofPiGroup	eq
o=mlusers for future mailserv feature	mail	eq
	mlsubListIdentifier	eq
	mlsubMail	eq

Should you decide to add further indexes on your own, instructions for adding indexes can be found in the Directory Server documentation.

Running the comm_dssetup.pl Script

This section covers the following topics:

- To Download comm_dssetup.pl
- To Install comm_dssetup.pl
- To Run the comm_dssetup.pl Script
- To Run the comm_dssetup.pl Script in Interactive Mode
- To Run the comm_dssetup.pl Script in Silent Mode
- Explanation of Options for Running comm_dssetup.pl Script in Silent Mode

To Download comm_dssetup.pl

1. Download the `comm_dssetup.pl` installer from the Oracle software delivery website, located at: http://edelivery.oracle.com/
   The `comm_dssetup.pl` installer is available as a download from any of the Unified Communications Suite media packs.
2. Copy the ZIP file to a temporary directory on your Directory Server hosts and extract the files.

To Install comm_dssetup.pl

1. Change to the directory where you extracted the ZIP file.
2. Launch the installer.

```
commpkg install
```

3. Choose the installation directory or accept the default.
4. From the item list, choose Comms DSsetup.

To Run the comm_dssetup.pl Script

1. On the server where Directory Server is installed, log in as or become superuser `root`.
2. Start Directory Server, if necessary.
3. Change to the directory where you installed or copied the `comm_dssetup.pl` script.
4. Run the `sbin/comm_dssetup.pl` script in either silent mode or in interactive mode.

For further steps, see To Run the comm_dssetup.pl Script in Interactive Mode or To Run the comm_dssetup.pl Script in Silent Mode.

To Run the comm_dssetup.pl Script in Interactive Mode

To run the `comm_dssetup.pl` script in interactive mode, run the script without any arguments and then enter your choices for the questions asked.

```
/usr/bin/perl comm_dssetup.pl
```

To Run the comm_dssetup.pl Script in Silent Mode

- `comm_dssetup.pl` Script Silent Mode Instructions
- `comm_dssetup.pl` Script Silent Mode Syntax

comm_dssetup.pl Script Silent Mode Instructions

To run the `comm_dssetup.pl` script in silent mode, issue the Perl command followed by a string of options using the syntax shown in comm_dssetup.pl Script Silent Mode Syntax. All of the option arguments are required.

The utility creates the following LDIF file and shell script to update the LDAP directory indexes and schema:

/var/tmp/dssetup_*timestamp*/dssetup.ldif

/var/tmp/dssetup_*timestamp*/dssetup.sh

Depending on the option values you pass in, the utility will either proceed to update the Directory Server by executing the new script, or not. If you have chosen not to proceed with the update, you can check the script and make any desired modifications before running the actual update at a later time.

comm_dssetup.pl Script Silent Mode Syntax

The following are all the options for running in the silent mode:

```
perl comm_dssetup.pl -i <yes|no> -R <yes|no> -c <DirectoryServerRoot> -d
<DirectoryInstance> -r <DCTreeSuffix> -u <UserGroupSuffix> -s <yes|no> -D
<DirectoryManagerDN> -j <DirectoryManagerPasswordFile> -b <yes|no> -t
<1|1.5|2> -m <yes|no> [-S <PathtoSchemaFiles>
```

Explanation of Options for Running comm_dssetup.pl Script in Silent Mode

Option and Argument	Description
-i yes|no	Answers the question: "Do you want to configure new indexes?" yes - Add new Directory Server indexes. no - Do not add indexes.
-R yes | no	Answers the question: "Do you want to reindex now?" The -m option must be yes also for this to take effect.
-c *DirectoryServerRoot*	Directory Server root path. For example: /var/opt/sun/directory
-d *DirectoryInstance*	Directory Server instance subdirectory under the Directory Server root path. For example: slapd-varrius
-r *DCTreeSuffix*	DC tree root suffix. (for Schema 1 and Schema 2 compatibility modes only) For example: o=internet
-u *UserGroupSuffix*	User and group root suffix. For example: o=usergroup
-s yes | no	Answers the question: "Do you want to update the schema?" yes - Update the schema. no - Do not update schema.
-D *DirectoryManagerDN*	Directory Manager Distinguished Name (DN). The value must be enclosed by double quotation marks (" ") to allow the comm_dssetup.pl script to interpret a value with a space correctly. For example: "cn=Directory Manager"
-j *DirectoryManagerPasswordFile*	File containing the Directory Manager DN password.
-b yes | no	Answers the question: "Will this directory server be used for users and groups?" yes - Use this directory to store both configuration and user group data. no - Use this directory to store only configuration data. This is only used for Messaging Server 6.2 or earlier
-t 1|1.5|2	Specifies the schema version.
-m yes | no	Answers the question: "Do you want to modify the directory server?" yes Modify the Directory Server without prompting the user. no Do not modify the Directory Server without prompting the user.
-S *PathtoSchemaFiles*	Path to the directory where the schema files are located. For example: ./schema

Manually Updating Schema Files

If for any reason, you have decided not to run the comm_dssetup.pl script generated script, the following directions allow you to manually update your schema files for Directory Server.

> **Note**
> If you update your LDAP directory schema manually and then later upgrade Calendar Server, you must manually update the LDAP server schema again. Calendar Server cannot automatically update the schema after it has previously been updated manually.

To Update Your LDAP Directory Manually

1. Install Calendar Server 7.
2. Stop Calendar Server, if it is running.
3. Stop Directory Server, if it is running.
4. Copy the `60iplanet-calendar.ldif` file to the following directory on the machine where your Directory Server is running:
   dir-svr-base`/slapd-`*hostname*`/config/schema` where *dir-svr-base* is the Directory Server installation directory and *hostname* identifies the machine.
5. If you want to index attributes, as the configuration program does, do it at this point.
   For a list of the attributes the configuration program indexes, see Attribute Indexes Created by the comm_dssetup.pl Script.
6. Restart the Directory Server.

If you receive object identifier (OID) errors, see Resolving Conflicting Calendar Server OID's in the LDAP Schema.

Resolving Conflicting Calendar Server OID's in the LDAP Schema

If your LDAP schema contains conflicting OID's, the Directory Server does not know which OID to use and returns an error message. For example, the following message indicates a conflicting OID for the `icsCalendarUser` object class:

```
[24/Apr/2004:23:45:28 -0700] dse -
The entry cn=schema in file 99user.ldif is invalid,
error code 20 (Type or value exists) - object class icscalendaruser:
The name does not match the OID.
Another object class is already using the name or OID.
[24/Apr/2004:23:45:28 -0700] dse -
Please edit the file to correct the reported problems
and then restart the server.
```

This problem can occur when you install Calendar Server and you also had an older Calendar Server release that dynamically updated your Directory Server `99user.ldif` file.
To resolve the conflicting OID's, perform the following two steps:

1. Edit the `99user.ldif` file and remove the older OID's. The following table lists the specific OID's that might cause problems.

Object Class	Old OID	New OID
icsCalendarUser	1.3.6.1.4.1.42.2.27.9.2.44	1.3.6.1.4.1.42.2.27.9.2.140
icsCalendarResource	1.3.6.1.4.1.42.2.27.9.2.45	1.3.6.1.4.1.42.2.27.9.2.141
icsCalendarDomain	1.3.6.1.4.1.42.2.27.9.2.4	1.3.6.1.4.1.42.2.27.9.2.149

2. After you edit the `99user.ldif` file, restart the Directory Server.

Chapter 4. Downgrading From Messaging Server 8.0

Downgrading From Messaging Server 8.0

If you upgrade using a coexistence migration strategy, you do not need to downgrade or back out a patch since you always have the system with the previous version of Messaging Server still running. Simply uninstall or decommission the newly installed version of Messaging Server on the new system and continue using the previous version on the old system. However, if you upgrade using a side-by-side or an in-place migration strategy, then you need to read the following information.

You cannot just back out the upgrade by using `commpkg uninstall` and then `commpkg install` from the previous release to reinstall the previous version. Instead, you must back up your Messaging Server data, back out the upgrade, then restore the Messaging Server data.

These downgrade instructions apply to both the in-place or side-by-side upgrade methods.

Topics:

- Before you Upgrade to Messaging Server 8.0
- Downgrading from Messaging Server 8.0 Without Using a ZFS Snapshot

Before you Upgrade to Messaging Server 8.0

Read this section before upgrading to Messaging Server 8.0 to understand how this release is different from previous releases.

- You cannot simply back out the Messaging Server 8.0 upgrade to a previous version once it is applied.
- Although the system does permit you to back out the upgrade (for example, by running `commpkg uninstall` and then `commpkg install` from the previous release to reinstall the previous version, afterwards Messaging Server services may not properly start. Additionally, the `stored` process may not start properly, and any mailbox accessed prior to backing out the upgrade may report that it is corrupted with an invalid format. Furthermore, even if you could manage to start Messaging Server services and manually fix the mailbox corruption, the mailbox owner flags (for example, seen and deleted flags) are all reset.
- Before upgrading to Messaging 8.0, make sure that you back up the Messaging Server data. If you do need to downgrade after upgrading to Messaging Server 8.0, you need to restore the Messaging Server data to their state prior to upgrading.
- Before upgrading to Messaging Server 8.0, it his highly recommended that you test it on a non-production system prior to actual deployment to production systems.
- Backing out from Messaging Server 8.0 is considered an avenue of last resort. If you need to downgrade, you must follow the steps described later in this information to return your system to a working state.
- You will need the previous version's software. For example, if you use the installer to upgrade from Messaging Server 7 Update 5, the installer removes the old software, and so to revert to that version, you would need the old product's installer to do so. Note that if you do a backup prior to downgrading, and restore from that backup, you do not lose messages that arrived since that backup when you restore.

Downgrading Using a ZFS Snapshot (Solaris Only)

To back out the upgrade on a host configured without a store such as an MTA host, an MMP host, or a Webmail host, run `commpkg uninstall` and then `commpkg install` from the previous release to reinstall the previous version. On a host configured with a Message Store that uses a ZFS file system, you can use the following procedure to back out the upgrade without having to do a full `imsbackup/imsrestore` thereby taking advantage of the near instantaneous ZFS snapshot and roll back capability.

High Level Overview

Create a ZFS snapshot of the message store data including the `mboxlist` database, index and message partitions **before** upgrading.

Once you upgrade, you can back out by:

- performing an incremental `imsbackup` of the message store since the snapshot time.
- using `commpkg uninstall` and then `commpkg install` from the previous release to reinstall the previous version.
- rolling back to the ZFS snapshot.
- restoring the incremental `imsbackup.`

Note however, that if you are backing out to a version prior to Messaging Server 7.0.5.29.0, those versions do not restore message flags from the incremental backup.

To Downgrade Using a ZFS Snapshot

1. Prior to upgrading, stop the services and create a ZFS snapshot of the Message Store. Note that in a subsequent step where a ZFS rollback is done to restore this snapshot, only the store area should be restored. In particular, you should not rollback the MTA queues. For additional information see the discussion of ZFS best practices in *Messaging Server System Administrator's Guide*. Make a note of the timestamp when you create the ZFS snapshot. We recommend using the timestamp in the name of the snapshot. The example below assumes that the store area is in the `rpool/export/comms-data` ZFS partition.

   ```
   stop-msg
   zfs list
   zfs snapshot rpool/export/comms-data@20150601232600
   ```

2. Upgrade and start services.

   ```
   commpkg upgrade
   start-msg
   ```

   If you decide for whatever reason to downgrade, note that this decision should not be taken lightly. This should only be done if there is no other recourse.

3. Stop services.

   ```
   stop-msg
   ```

4. Start Message Store services.

   ```
   start-msg store
   ```

5. Do an incremental `imsbackup` from the time the ZFS snapshot was taken in Step 1. (timestamp 2015-Jun-01 11:26 pm)

```
imsbackup -x -v -f - -d 20150601:232600 / > /var/tmp/backup
```

Note: It is better if the incremental backup is relatively small.

6. Stop services.

```
stop-msg
```

It would seem prudent to do another ZFS snapshot prior to starting the downgrade, but ZFS snapshots do not allow you to rollback to more than the previous snapshot.

7. Uninstall the Messaging Server

```
commpkg uninstall
```

8. Reinstall the previous Messaging Server version by running its installer.

```
commpkg install
```

9. Roll back to the ZFS snapshot you created previously.

```
zfs rollback rpool/export/comms-data@20150601232600
```

10. Start the message store services.

```
start-msg store
```

11. Restore the backup you made previously using `imsbackup` by running `imsrestore` with the -E switch.

```
imsrestore -E -f /var/tmp/backup
```

12. Start services.

```
start-msg
```

Downgrading from Messaging Server 8.0 Without Using a ZFS Snapshot

Use this procedure if you have upgraded to Messaging Server 8.0 and need to return to previous version.

1. Prior to downgrading, perform a full backup of the message store by using the `imsbackup` command.
   For example:

```
stop-msg
start-msg store
imsbackup -v -f - / > backup
```

2. Uninstall the Messaging Server

```
commpkg uninstall
```

3. Reinstall the previous Messaging Server version by running its installer.

   ```
   commpkg install
   ```

4. Move the `store` directory to a different location.
   For example:

   ```
   mv data/store data/store.old
   ```

5. Start the message store to perform the restore.

   ```
   start-msg store
   ```

6. Perform the restore.

   ```
   imsrestore -f backup
   ```

7. Start Messaging Server.
   For example:

   ```
   start-msg
   ```

Chapter 5. Installation Scenarios - Messaging Server 8.0

Installation Scenarios - Oracle Communications Messaging Server 8.0

This page lists the Messaging Server 8.0 installation scenarios.

- Installation Scenario - Messaging Server 8.0 Message Store
- Installation Scenario - Messaging Server 8.0 Message Transfer Agent
- Installation Scenario - Messaging Server 8.0 Messaging Multiplexor
- Installation Scenario - Messaging Server 8.0 Webmail Server

Installation Scenario - Messaging Server 8.0 Message Store

Installation Scenario - Messaging Server 8.0 Message Store

Beginning with Messaging Server 7 Update 5, you must decide if you want to use Unified Configuration or legacy configuration. Unified Configuration is an improved, streamlined process to configure and administer Messaging Server. Unlike in legacy configurations (Messaging Server 7 Update 4 and prior releases), Unified Configuration uses validation to verify configuration accuracy, and employs a single tool to configure the entire Messaging Server configuration (with a few exceptions). For more information, see the overview of Messaging Server Unified Configuration in *Messaging Server Unified Configuration System Administrator's Guide*.

Topics:

- Installation Assumptions
- Downloading the Messaging Server Software
- Installing and Configuring the Messaging Store

Installation Assumptions

This scenario describes how to install the Messaging Server back-end message store using the following assumptions:

- Oracle Directory Server Enterprise Edition (Directory Server) is already deployed at your site. Prior to installing and configuring Messaging Server, you must also prepare the Directory Server LDAP schema by running the `comm_dssetup.pl` script. This script, which is provided as part of the Messaging Server media pack, adds the necessary Communications Suite schema to the LDAP. See Preparing Directory Server for more information.
- You are deploying Messaging Server on multiple hosts or Solaris zones.
- This Messaging Server back-end message store is one functional component of your multi-host deployment.
- You are installing the message store on a separate host or Solaris zone. You are not installing the message store with other Communications Suite products on the same host.
- If you are distributing multiple partitions of the message store across several hosts or zones, you can follow these instructions for each host on which you install store partitions.

Downloading the Messaging Server Software

1. Download the media pack for Oracle Communications Messaging Server from the Oracle software delivery website, located at:
   http://edelivery.oracle.com/
   The Messaging Server media pack contains installers for Messaging Server, the `comm_dssetup` script, and other software that you are licensed to use.
2. Copy the Messaging Server ZIP file to a temporary directory on your Messaging Server hosts and extract the files.

Installing and Configuring the Messaging Store

Before Installing the Message Store

1. Ensure that DNS is running and configured properly.
   For details, see the topic on DNS configuration in *Unified Communications Suite Installation and Configuration Guide*.
2. Review the recommended message store file systems in *Unified Communications Suite*

Installation and Configuration Guide.

3. Make sure you do not configure conflicting port numbers on a host when various components are running on a single machine.
   For a list of port numbers used by Messaging Server, see the topic on default port numbers in *Unified Communications Suite Installation and Configuration Guide*.

Preparing Directory Server

Prior to installing and configuring Messaging Server, you must also prepare the Directory Server LDAP schema by running the `comm_dssetup.pl` script. This script, which is included as a separate installable component of the Messaging Server media pack that you previously downloaded, adds the necessary schema to the LDAP. For Messaging Server 8.0, you must use `comm_dssetup.pl` 6.4.0.27.0 or greater.

1. Copy the Comms DSsetup ZIP file to a temporary directory on your Directory Server hosts and extract the files.
2. Install and run the `comm_dssetup.pl` script.
   For more information, see Running the comm_dssetup.pl Script.

   > **Note**
   > You can use either LDAP Schema 2 or Schema 1.

3. If necessary, provision users in the Directory Server.
   If Directory Server is already installed at your site, users have already been provisioned. If you have just installed Directory Server at your site, then you need to provision users. For information about provisioning users and schema, see *Unified Communications Suite Schema Reference*.

To Install the Message Store

1. On the message store host, log in as or become the superuser (`root`).
2. Change to the directory in which you extracted the Messaging Server ZIP file.
3. Launch the installer.

   ```
   commpkg install
   ```

4. Choose the installation directory or accept the default.
5. From the item list, choose Messaging Server.
   When the installation is complete, continue with the To Configure the Message Store section.

To Get GlassFish Message Queue

You can get GlassFish Message Queue in one of the following ways:

- Get the Indexing and Search Service standalone installer and use it to install GlassFish Message Queue.
- Get GlassFish and install the embedded Message Queue.

To Configure the Message Store

You must configure Messaging Server to complete the installation. You use the Messaging Server configuration command-line script, `configure`, to perform this initial runtime configuration. For detailed instructions on performing an initial configuration, see Messaging Server 8.0 Initial Configuration.

1. Use the following worksheet to gather configuration information for the message store:
   Configuration Worksheets - Messaging Server 8.0

2. On the message store host, log in as or become the superuser (`root`).
3. Change to the *MessagingServer_home*`/sbin` directory:
   The default *MessagingServer_home* installation directory is `/opt/sun/comms/messaging64`.
4. To configure a legacy configuration, run the `configure` command.
   For more information on the `configure` options, see To Run the Configure Program.
5. To configure a Unified Configuration, run the `configure --xml` command.
   For more information on options to the `configure --xml` command, see *Messaging Server Unified Configuration System Administrator's Guide*.
6. If you are not also using the Webmail server on this message store, disable it.
   - In legacy configuration, run this command:

     ```
     configutil -o service.http.enable -v 0
     ```

   - In Unified Configuration, run this command:

     ```
     msconfig set http.enable 0
     ```

7. If you are configuring LMTP, see *Messaging Server Unified Configuration System Administrator's Guide*.
8. If you are not using the MTA, disable it.
   - In legacy configuration, run this command:

     ```
     configutil -o local.imta.enable -v 0
     ```

   - In Unified Configuration, run this command:

     ```
     msconfig set mta.enable 0
     ```

Installation Scenario - Messaging Server 8.0 Message Transfer Agent

Installation Scenario - Messaging Server 8.0 Message Transfer Agent

Beginning with Messaging Server 7 Update 5, you need to decide if you want to use Unified Configuration or legacy configuration. Unified Configuration is an improved, streamlined process to configure and administer Messaging Server. Unlike in legacy configurations (Messaging Server 7 Update 4 and prior releases), Unified Configuration uses validation to verify configuration accuracy, and employs a single tool to configure the entire Messaging Server configuration (with a few exceptions). For more information, see the overview of Messaging Server Unified Configuration in *Messaging Server Unified Configuration System Administrator's Guide*.

Topics:

- Installation Assumptions
- Downloading the Messaging Server Software
- Installing and Configuring the MTA

Installation Assumptions

This scenario describes how to install the Messaging Server Message Transfer Agent (MTA) on a separate host using the following assumptions:

- Oracle Directory Server Enterprise Edition (Directory Server) is already deployed at your site. Prior to installing and configuring Messaging Server, you must also prepare the Directory Server LDAP schema by running the `comm_dssetup.pl` script. This script, which is provided as part of the Messaging Server media pack, adds the necessary Communications Suite schema to the LDAP. See Preparing Directory Server for more information.
- You are deploying Messaging Server on multiple hosts or Solaris zones.
- This MTA relay in and MTA relay out is one functional component of your multi-host deployment.
- You are installing the MTA on a separate host or Solaris zone. You are not installing the MTA with other Communications Suite products on the same host.
- If you are distributing multiple instances of the MTA across several hosts or zones, you can follow these instructions for each host on which you install the MTA.

Downloading the Messaging Server Software

1. Download the media pack for Oracle Communications Messaging Server from the Oracle software delivery website, located at:
   http://edelivery.oracle.com/
   The Messaging Server media pack contains installers for Messaging Server, the `comm_dssetup` script, and other software that you are licensed to use.
2. Copy the Messaging Server ZIP file to a temporary directory on your Messaging Server hosts and extract the files.

Installing and Configuring the MTA

Before Installing the MTA

1. Ensure that DNS is running and configured properly.
   For details, see the topic on DNS configuration in *Unified Communications Suite Installation and Configuration Guide*.

2. Make sure you do not configure conflicting port numbers on a host when various components are running on a single machine.
   For a list of port numbers used by Messaging Server, see the topic on default port numbers in *Unified Communications Suite Installation and Configuration Guide*.

Preparing Directory Server

Prior to installing and configuring Messaging Server, you must also prepare the Directory Server LDAP schema by running the `comm_dssetup.pl` script. This script, which is included as a separate installable component of the Messaging Server media pack that you previously downloaded, adds the necessary schema to the LDAP. For Messaging Server 8.0, you must use `comm_dssetup.pl` 6.4.0.27.0 or greater.

1. Copy the Comms DSsetup ZIP file to a temporary directory on your Directory Server hosts and extract the files.
2. Install and run the `comm_dssetup.pl` script.
   For more information, see Running the comm_dssetup.pl Script.

   > **Note**
   > You can use either LDAP Schema 2 or Schema 1.

3. If necessary, provision users in the Directory Server.
   If Directory Server is already installed at your site, users have already been provisioned. If you have just installed Directory Server at your site, then you need to provision users. For information about provisioning users and schema, see *Unified Communications Suite Schema Reference*.

To Install the MTA

1. On the MTA host, log in as or become the superuser (`root`).
2. Change to the directory in which you extracted the Messaging Server ZIP file.
3. Launch the installer.

   ```
   commpkg install
   ```

4. Choose the installation directory or accept the default.
5. From the item list, choose Messaging Server.
   When the installation is complete, continue with the To Configure the MTA section.

To Get GlassFish Message Queue

You can get GlassFish Message Queue in one of the following ways:

- Get the Indexing and Search Service standalone installer and use it to install GlassFish Message Queue.
- Get GlassFish and install the embedded Message Queue.

To Configure the MTA

You must configure Messaging Server to complete the installation. You use the Messaging Server configuration command-line script, `configure`, to perform this initial runtime configuration. For detailed instructions on performing an initial configuration, see Messaging Server 8.0 Initial Configuration.

1. Use the following worksheet to gather configuration information for the MTA: Configuration Worksheets - Messaging Server 8.0
2. On the MTA host, log in as or become the superuser (`root`).

3. Change to the *MessagingServer_home*/`sbin` directory:
   The default installation directory is `/opt/sun/comms/messaging64`.
   - To configure a legacy configuration, run the `configure` command.
     For more information on options to the `configure` command, see To Run the Configure Program.
   - To configure a Unified Configuration, run the `configure --xml` command.
     For more information on options to the `configure --xml` command, see *Messaging Server Unified Configuration System Administrator's Guide*.
4. Disable the message store and Webmail server.
   - In legacy configuration, run these commands:

     ```
     configutil -o local.store.enable -v 0
     configutil -o service.http.enable -v 0
     ```

   - In Unified Configuration, run these commands:

     ```
     msconfig set store.enable 0
     msconfig set http.enable 0
     ```

5. Configure the relay for the kind of traffic you are dealing with and the kind of traffic shaping you need.
   For example, if your inbound relay needs to use LMTP, configure your deployment accordingly.

Installation Scenario - Messaging Server 8.0 Messaging Multiplexor

Installation Scenario - Messaging Server 8.0 Messaging Multiplexor

Beginning with Messaging Server 7 Update 5, you need to decide if you want to use Unified Configuration or legacy configuration. Unified Configuration is an improved, streamlined process to configure and administer Messaging Server. Unlike in legacy configurations (Messaging Server 7 Update 4 and prior releases), Unified Configuration uses validation to verify configuration accuracy, and employs a single tool to configure the entire Messaging Server configuration (with a few exceptions). For more information, see the overview of Messaging Server Unified Configuration in *Messaging Server Unified Configuration System Administrator's Guide*.

Topics:

- Installation Assumptions
- Downloading the Messaging Server Software
- Installing and Configuring the MMP

Installation Assumptions

This scenario describes how to install the Messaging Multiplexor (MMP) front-end host using the following assumptions:

- Oracle Directory Server Enterprise Edition (Directory Server) is already deployed at your site. Prior to installing and configuring Messaging Server, you must also prepare the Directory Server LDAP schema by running the `comm_dssetup.pl` script. This script, which is provided as part of the Messaging Server media pack, adds the necessary Communications Suite schema to the LDAP. See Preparing Directory Server for more information.
- You are deploying Messaging Server on multiple hosts or Solaris zones.
- This MMP front-end host is one functional component of your multi-host deployment.
- You are installing the MMP on a separate host or Solaris zone. You are not installing the MMP with other Communication Suite products on the same host.
- If you are distributing multiple instances of the MMP across several hosts or zones, you can follow these instructions for each host on which you install the MMP.
- You are installing only the MMP front end; you are not installing message store or SMTP functions.

Downloading the Messaging Server Software

1. Download the media pack for Oracle Communications Messaging Server from the Oracle software delivery website, located at:
   http://edelivery.oracle.com/
   The Messaging Server media pack contains installers for Messaging Server, the `comm_dssetup` script, and other software that you are licensed to use.
2. Copy the Messaging Server ZIP file to a temporary directory on your Messaging Server hosts and extract the files.

Installing and Configuring the MMP

Before Installing the MMP

1. Ensure that DNS is running and configured properly.

For details, see the topic on DNS configuration in *Unified Communications Suite Installation and Configuration Guide*.

2. Make sure you do not configure conflicting port numbers on a host when various components are running on a single machine.
   For a list of port numbers used by Messaging Server, see the topic on default port numbers in *Unified Communications Suite Installation and Configuration Guide*.

Preparing Directory Server

Prior to installing and configuring Messaging Server, you must also prepare the Directory Server LDAP schema by running the `comm_dssetup.pl` script. This script, which is included as a separate installable component of the Messaging Server media pack that you previously downloaded, adds the necessary Communications Suite schema to the LDAP. For Messaging Server 8.0, you must use `comm_dssetup.pl` 6.4.0.27.0 or greater.

1. Copy the Comms DSsetup ZIP file to a temporary directory on your Directory Server hosts and extract the files.
2. Install and run the `comm_dssetup.pl` script.
   For more information, see Running the comm_dssetup.pl Script.

   > **Note**
   > You can use either LDAP Schema 2 or Schema 1.

3. If necessary, provision users in the Directory Server.
   If Directory Server is already installed at your site, users have already been provisioned. If you have just installed Directory Server at your site, then you need to provision users. For information about provisioning users and schema, see *Unified Communications Suite Schema Reference*.

To Install the MMP

1. On the MMP host, log in as or become the superuser (`root`).
2. Change to the directory in which you extracted the Messaging Server ZIP file.
3. Launch the installer.

   ```
   commpkg install
   ```

4. Choose the installation directory or accept the default.
5. From the item list, choose Messaging Server.
   When the installation is complete, continue with the To Configure the MMP section.

To Get GlassFish Message Queue

You can get GlassFish Message Queue in one of the following ways:

- Get the Indexing and Search Service standalone installer and use it to install GlassFish Message Queue.
- Get GlassFish and install the embedded Message Queue.

To Configure the MMP

You must configure Messaging Server to complete the installation. You use the Messaging Server configuration command-line script, `configure`, to perform this initial runtime configuration. For detailed instructions on performing an initial configuration, see Messaging Server 8.0 Initial Configuration.

1. Use the following worksheet to gather configuration information for the MMP: Configuration

Worksheets - Messaging Server 8.0

2. On the MMP host, log in as or become the superuser (`root`).
3. Change to the *MessagingServer_home*/`sbin` directory:
   The default installation directory is `/opt/sun/comms/messaging64`.
4. To configure a legacy configuration, run the `configure` command.
    - To configure a legacy configuration, run the `configure` command.
      For more information on options to the `configure` command, see To Run the Configure Program.
    - To configure a Unified Configuration, run the `configure --xml` command.
      For more information on options to the `configure --xml` command, see *Messaging Server Unified Configuration System Administrator's Guide*.
5. Enable the MMP and disable other product components.
    - In legacy configuration, run these commands:

      ```
      configutil -o local.mmp.enable -v 1
      configutil -o local.store.enable -v 0
      configutil -o local.imta.enable -v 0
      configutil -o service.http.enable -v 0
      ```

    - In Unified Configuration, run these commands:

      ```
      msconfig set mmp.enable 1
      msconfig set store.enable 0
      msconfig set mta.enable 0
      msconfig set http.enable 0
      ```

Installation Scenario - Messaging Server 8.0 Webmail Server

Installation Scenario - Messaging Server 8.0 Webmail Server

Beginning with Messaging Server 7 Update 5, you need to decide if you want to use Unified Configuration or legacy configuration. Unified Configuration is an improved, streamlined process to configure and administer Messaging Server. Unlike in legacy configurations (Messaging Server 7 Update 4 and prior releases), Unified Configuration uses validation to verify configuration accuracy, and employs a single tool to configure the entire Messaging Server configuration (with a few exceptions). For more information, see the overview of Messaging Server Unified Configuration in *Messaging Server Unified Configuration System Administrator's Guide*.

Topics:

- Installation Assumptions
- Downloading the Messaging Server Software
- Installing and Configuring Webmail Server
- Configuring Webmail Server Examples

Installation Assumptions

This scenario describes how to install the Messaging Server Webmail server (`mshttpd`) on a separate host. The Webmail server acts as a front end that handles the HTTP protocol retrieval of messages from the message store. This component is used by Convergence to provide web-based access to end users.

This scenario describes how to install the Messaging Server Webmail server using the following assumptions:

- Oracle Directory Server Enterprise Edition (Directory Server) is already deployed at your site. Prior to installing and configuring Messaging Server, you must also prepare the Directory Server LDAP schema by running the `comm_dssetup.pl` script. This script, which is provided as part of the Messaging Server media pack, adds the necessary Communications Suite schema to the LDAP. See Preparing Directory Server for more information.
- You are deploying Messaging Server on multiple hosts or Solaris zones.
- This Webmail server is one functional component of your multi-host deployment.
- You are installing the Webmail server on a separate host. You are not installing the Webmail server with other Communication Suite products on the same host.
- If you are distributing multiple Webmail servers across several hosts, you can follow these instructions for each host on which you install the Webmail server.
- You are installing only the Webmail server front end; you are not installing message store or SMTP functions.

Downloading the Messaging Server Software

1. Download the media pack for Oracle Communications Messaging Server from the Oracle software delivery website, located at:
   http://edelivery.oracle.com/
   The Messaging Server media pack contains installers for Messaging Server, the `comm_dssetup` script, and other software that you are licensed to use.
2. Copy the Messaging Server ZIP file to a temporary directory on your Messaging Server hosts and extract the files.

Installing and Configuring Webmail Server

Before Installing the Webmail Server

1. Ensure that DNS is running and configured properly.
   For details, see the topic on DNS configuration in *Unified Communications Suite Installation and Configuration Guide*.
2. Make sure you do not configure conflicting port numbers on a host when various components are running on a single machine.
   For a list of port numbers used by Messaging Server, see the topic on default port numbers in *Unified Communications Suite Installation and Configuration Guide*.

Preparing Directory Server

Prior to installing and configuring Messaging Server, you must also prepare the Directory Server LDAP schema by running the `comm_dssetup.pl` script. This script, which is included as a separate installable component of the Messaging Server media pack that you previously downloaded, adds the necessary schema to the LDAP. For Messaging Server 8.0, you must use `comm_dssetup.pl` 6.4.0.27.0 or greater.

1. Copy the Comms DSsetup ZIP file to a temporary directory on your Directory Server hosts and extract the files.
2. Install and run the `comm_dssetup.pl` script.
   For more information, see Running the comm_dssetup.pl Script.

   > **Note**
   > You can use either LDAP Schema 2 or Schema 1.

3. If necessary, provision users in the Directory Server.
   If Directory Server is already installed at your site, users have already been provisioned. If you have just installed Directory Server at your site, then you need to provision users. For information about provisioning users and schema, see *Unified Communications Suite Schema Reference*.

To Install the Webmail Server

1. On the Webmail server host, log in as or become the superuser (`root`).
2. Change to the directory in which you extracted the Messaging Server ZIP file.
3. Launch the installer.

   ```
   commpkg install
   ```

4. Choose the installation directory or accept the default.
5. From the item list, choose Messaging Server.
   When the installation is complete, continue with the To Configure the Message Store section.

To Get GlassFish Message Queue

You can get GlassFish Message Queue in one of the following ways:

- Get the Indexing and Search Service standalone installer and use it to install GlassFish Message Queue.
- Get GlassFish and install the embedded Message Queue.

To Configure the Webmail Server

You must configure Messaging Server to complete the installation. You use the Messaging Server configuration command-line script, `configure`, to perform this initial runtime configuration. For detailed instructions on performing an initial configuration, see Messaging Server 8.0 Initial Configuration.

1. Use the following worksheet to gather configuration information for the message store: Configuration Worksheets - Messaging Server 8.0
2. On the Webmail server host, log in as or become the superuser (`root`).
3. Change to the *MessagingServer_home/*`sbin` directory:
   The default installation directory is `/opt/sun/comms/messaging64`.
   - To configure a legacy configuration, run the `configure` command.
     For more information on options to the `configure` command, see To Run the Configure Program.
   - To configure a Unified Configuration, run the `configure --xml` command.
     For more information on options to the `configure --xml` command, see *Messaging Server Unified Configuration System Administrator's Guide*.
4. Disable the Message Store and MTA on the WebMail server host.
   - For legacy configuration:

     ```
     configutil -o local.store.enable -v 0
     configutil -o local.imta.enable -v 0
     ```

   - For Unified Configuration:

     ```
     msconfig set store.enable 0
     msconfig set mta.enable 0
     ```

5. (Optional) Set the following options.
   If you want to use a different store administrator or a non-standard IMAP port, use the following options for the back-end IMAP server(s):

Unified Configuration Option	Legacy Configuration Option	Description
`base.proxyadmin`	`local.service.proxy.admin`	Default back-end store administrator login name. (Restart of HTTP service required and restart of IMAP service required.) Syntax: string Default: `admin`
`base.proxyadminpass`	`local.service.proxy.adminpass`	Default store administrator password. (Restart of HTTP service required and restart of IMAP service required.) Syntax: string Default: *<admin.password>*
`base.proxyimapport`	`local.service.proxy.imapport`	Default IMAP port number for backend store servers. (Restart of HTTP service required and restart of IMAP service required.) Syntax: integer Default: 143

The Webmail server can communicate with multiple back-end IMAP servers. If the IMAP servers use different values for these options, you must set individual values for each host, as follows:

Unified Configuration Option	Legacy Configuration

`proxy:`*hostname*`.admin`
`proxy:`*hostname*`.adminpass`
`proxy.`*hostname*`.imapport`
	`local.service.proxy.admin.`*hostname*
	`local.service.proxy.adminpass.`*hostname*
	`local.service.imapport.`*hostname*

where *hostname* is the name of the host on which each back-end IMAP server is running.

Note
In general in Unified Configuration, for proxy-related options there should me two scopes for the same option:

- `base.`*option* is the global scope.
- `proxy:`*hostname.option* is the host-specific scope.

Currently, an error in Unified Configuration causes the same option to have two different names depending on the scope. Thus, `base.proxyimapport` is equivalent to `proxy:`*hostname*`.imapport`, `base.proxyimapport` is equivalent to `proxy:`*hostname*`.imapport`, and `base.proxyimapadminpass` is equivalent to `proxy:`*hostname*`.imapadminpass`. In addition, there is no host-specific form for `base.proxyimapssl`. It is a single global setting.

Configuring Webmail Server Examples

Topics in this section:

- Legacy Configuration
- Unified Configuration

Legacy Configuration

For one back-end IMAP server:

```
configutil -o local.service.proxy.admin -v myadmin
configutil -o local.service.proxy.adminpass -v <password>
configutil -o local.service.proxy.imapport -v 143
```

For multiple back-end IMAP servers:

```
configutil -o local.service.proxy.admin.host1.siroe.com -v admin1
configutil -o local.service.proxy.adminpass.host1.siroe.com -v <password>
configutil -o local.service.proxy.imapport.host1.siroe.com -v 143

configutil -o local.service.proxy.admin.host2.siroe.com -v admin2
configutil -o local.service.proxy.adminpass.host2.siroe.com -v <password>
configutil -o local.service.proxy.imapport.host2.siroe.com -v 143
```

Unified Configuration

For one back-end IMAP server:

```
msconfig set base.proxyadmin -myadmin
msconfig set base.proxyadminpass <password>
msconfig set base.proxyimapport -143
```

Chapter 6. Messaging Server 8.0 Initial Configuration

Oracle Communications Messaging Server 8.0 Initial Configuration

After you install the Messaging Server software, you must configure Messaging Server to complete the installation. You perform this initial runtime configuration by using the Messaging Server configuration program, `configure`.

This information assumes that you have read *Unified Communications Suite Deployment Planning Guide* and installed Messaging Server software. Performing the following tasks results in a functioning Messaging Server. You still want to customize your deployment as well as provision and perhaps migrate users and groups. Provisioning is described in *Delegated Administrator Administration Guide*.

Topics:

About Messaging Server Unified Configuration

Starting with version 7.0.5.29.0, Messaging Server introduces the capability to create a Unified Configuration. Unified Configuration provides an improved, streamlined process to configure and administer Messaging Server. Unlike in legacy configurations (Messaging Server 7 Update 4 and prior releases), Unified Configuration uses validation to verify configuration accuracy, and employs a single tool to configure the entire Messaging Server configuration (with a few exceptions). Thus, using Unified Configuration simplifies administration and reduces configuration mistakes.

When you perform a fresh Messaging Server installation, you can decide to configure it for Unified Configuration. It is not a requirement to use Unified Configuration with Messaging Server, however, Unified Configuration provides a number of benefits over legacy configuration. If you decide to not use Unified Configuration, rerun the `configure` command without `--xml` option to create a legacy configuration, then recreate any configuration changes you made while running under Unified Configuration.

To learn more about Unified Configuration, see the overview of Messaging Server Unified Configuration in *Messaging Server Unified Configuration System Administrator's Guide*.

Prerequisites for Configuring Messaging Server

Before running the `configure` program, you must:

- Install and configure the Directory Server.
- Run the `comm_dssetup.pl` program. See Communications Suite Directory Server Setup Script (comm_dssetup.pl).
- Record your Directory installation and configuration parameters in the checklists supplied in Installation Worksheets - Directory Server.

Messaging Server Configuration Checklist

Before you run the `configure program`, record your parameter choices in Configuration Worksheets - Messaging Server. To answer certain questions, refer to your Directory Server installation checklists in Installation Worksheets - Directory Server.

High-level Overview of Configuring Messaging Server

Performing an initial run-time configuration of Messaging Server involves the following high-level steps:

1. Creating a Unix system user and group for Messaging Server
2. Checking that DNS is properly configured
3. Preparing Directory Server for Messaging Server configuration by running the `comm_dssetup.pl` script
4. Creating the initial Messaging Server runtime configuration by running the `configure` command

Additionally, other steps to perform include the following:

1. Installing tools to provision Messaging Server
2. Modifying SMTP relay blocking configuration
3. Enabling Messaging Server startup after a reboot
4. Becoming familiar with best practices for performance tuning

The following sections describe in detail how to configure Messaging Server.

Creating UNIX System Users and Groups

System users run specific server processes, and privileges need to be given to these users so that they have appropriate permissions for the processes they are running.

Set up a system user account and group for all servers (for example, Messaging Server), and set permissions for the directories and files owned by that user.

> **Note**
> For security reasons, in some deployments it might be desirable to have different system administrators for different servers. This is done by creating different system users and groups per server. For example, the system user for Messaging Server would be different from the system user for Web Server, and system administrators administering Messaging Server would not be able to administer the Web Server.

To Create UNIX System Users and Groups

Creating UNIX system users and groups is optional. The `configure` initial configuration script does this if necessary.

1. Log in as `root`.
2. Create a group name for server processes to which your system users belong.
   For example:

   ```
   groupadd mail
   ```

3. Create a user name for system processes and associate it with the group name you just created. In addition, set the password for that user.
   For example:

   ```
   useradd -g mail mailsrv
   ```

   The `useradd` and `usermod` commands are located in the `/usr/sbin` directory. See UNIX man pages for more information.

4. You might also need to check the `/etc/group` and `/etc/passwd` files to be sure that the user has been added to the system group that you created.

   > **Note**
   > Should you decide not to set up UNIX system users and groups prior to installing Messaging Server, you are able to specify them when you run the configuration script. However, if the user name for server processes already exists, then the primary group for that user name is used, and the configuration script does not prompt for the option.

Checking the DNS Configuration

Check that DNS is running and configured properly for the Messaging Server host. The following example is for a host running Solaris 10 OS. The configuration is slightly different for a host running Solaris 11 OS.

1. Ensure that DNS is properly configured and that it is clearly specified how to route to hosts that are not on the local subnet.
   - The `/etc/defaultrouter` file should contain the IP address of the gateway system. This address must be on a local subnet.
   - The `/etc/resolv.conf` file exists and contains the proper entries for reachable DNS servers and domain suffixes.
   - In the `/etc/nsswitch.conf` file, the `hosts:` and `ipnodes:` line has the `files`, `dns` and `nis` keywords added. The keyword `files` must precede `dns` and `nis`. So if the lines look like this:

     ```
     hosts: nis dns files
     ipnodes: nis dns files
     ```

     They should be changed to this:

```
hosts: files nis dns
ipnodes: files nis dns
```

2. Make sure that the FQDN is the first host name specified after the IP address in the `/etc/hosts` file.
   If your Internet host table in your `/etc/hosts` file looks like this:

   ```
   123.456.78.910 budgie.west.sesta.com
   123.456.78.910 budgie loghost mailhost
   ```

   Change it so that there is only one line for the IP address of the host. Be sure the first host name is a fully qualified domain name. For example:

   ```
   123.456.78.910 budgie.west.sesta.com budgie loghost mailhost
   ```

   - You can verify that the lines are read correctly by running the following commands:

     ```
     # getent hosts <ip_address>
     # getent ipnodes <ip_address>
     ```

     If the lines are read correctly, you should see the IP address followed by the FQDN and then the other values.
     For example:

     ```
     # getent hosts 192.18.126.103
     192.18.126.103 budgie.west.sesta.com budgie loghost mailhost
     ```

Preparing Directory Server for Messaging Server Configuration

For more information on directory preparation and the directory preparation script `comm_dssetup.pl`, see Communications Suite Directory Server Setup Script (comm_dssetup.pl). The `comm_dssetup.pl` script prepares the Directory Server by setting up new schema, index, and data in your Directory Server. Run `comm_dssetup.pl` before installing or upgrading any software that is dependent on the Directory Server (such as Messaging Server, Calendar Server, Convergence, and so on).

> **Note**
> Always run the latest version of `comm_dssetup.pl` if you are upgrading any of the component products that depend on Directory Server.

Creating the Initial Messaging Server Runtime Configuration

The `configure` program provides a configuration to get your Messaging Server up and running. It is meant to create an initial runtime configuration to set up a generic functional Messaging Server configuration. Thus it gives you a base working configuration from which you can make your specific customizations. The program is only meant to be run once. Subsequent running of this program overwrites the existing configuration. To modify your initial runtime configuration, use the configuration utilities described here and in *Messaging Server Administration Reference*.

The `configure` command detects mismatches in certain critical LDAP attributes when performing second and subsequent initial configurations using the same LDAP server. The critical attributes are:

- default domain: `inetDomainBaseDN`, `preferredMailHost`, and `sunPreferredDomain`
- admin user: userPassword, mailHost, and mail

The admin's `userPassword` must match unless the `--novalidate` or `--noldap` options are used with `configure` (in which case the new value will replace the old one when the LDIF generated by `configure` is applied). In interactive mode, the admin may select whether to preserve or replace the other attributes. The default behavior is replace (as with previous versions), but the new `--preserveCritical` option changes the default behavior to preserve. If a state file is used, the default behavior is applied to all attributes except `userPassword`.

To Run the Configure Program

1. Invoke the Messaging Server initial runtime `configure` command.
    - To configure a legacy configuration, run *msg-svr-base/*`sbin/configure --noxml`
    - To configure a Unified Configuration, run *msg-svr-base/*`sbin/configure`
      For more information on deciding to use Unified Configuration, see the overview of Messaging Server Unified Configuration in *Messaging Server Unified Configuration System Administrator's Guide*.
      The following table describes options you can set with the `configure` program:

Option	Description
`--debug`	Provides general debug information primarily for LDAP operations.
`--help`	Displays help
`--ignoreSendmail`	Keeps sendmail enabled after configuration. In other words, does not disable sendmail after configuration.
`--ldapport` [*ldapport*]	Specifies an LDAP port other than the default port 389.
`--ldif`	Causes configure to run without modifying the directory and instead generate an ldif file (*msg-svr-base* `/data/install/configure.ldif`) which the admin can apply to the directory after initial configuration. This is needed if the person doing the installation does not have directory admin rights.
`--noldap`	Runs without LDAP present (statefile only)
`--novalidate`	Skips most validation of user input.
`--noxml`	Generates legacy configuration (does not use XML-based Unified Configuration); can also be used to replace a Unified Configuration with a freshly generated legacy configuration (fresh installation of Messaging Server, not an upgrade where the `configtoxml` command was run).
`--preserveCritical`	Changes the default behavior from replace to preserve.
`--saveState` [*statefile*]	Specifies a location other than the default location (mentioned below) to save a state file.
`--ssl` [*ssl*]	Requires SSL when configuring LDAP.
`--state` [*statefile*]	Uses a silent installation file. See To Perform a Silent Installation.
`--version, --V`	Displays product version.
`--xml`	Generates Unified Configuration (XML).

After running the command, the welcome text appears.

2. Select the directory where you want to store the Messaging Server configuration and data files. Symbolic links are created under the *msg-svr-base* directory to the configuration and data directory. For more information on these symbolic links, see Post-Installation Directory Layout.
   Make sure you have large enough disk space set aside for these files.
   The "Overwrite the existing configuration" prompt appears if you have an existing configuration.
   a. If you do receive the "Overwrite" message, to accept the default of yes, press Enter.
   b. Otherwise, type **n** to enter a different directory path.
3. Select the user name for server processes.
   To accept the default user name `mailsrv`, press Enter. Otherwise, type the user name for the server processes.
4. Select the group name for server processes.
   To accept the default group name `mail`, press Enter. Otherwise, type the group name for the server processes. This question appears only if the UNIX user name has not yet been created.
5. Select the fully-qualified local host name.

This is the machine on which Messaging Server runs. When you installed the server, you might have specified the physical host name. However, if you are installing a cluster environment, use the logical host name. Here is the chance to change what you originally specified.

6. Type the default mail domain.
7. Select the host name for the LDAP Directory Server.
8. Select the LDAP administrator login.
   The Directory Manager has overall administrator privileges on the Directory Server and all servers (for example, Messaging Server) that make use of the Directory Server, and has full administration access to all entries in the Directory Server. The default and recommended Distinguished Name (DN) is `cn=Directory Manager` and is set during Directory Server configuration
   If you are installing against a replicated Directory Server instance, you must specify the credentials of the replica, not the master directory.
9. Type the LDAP administrator password.
   Messages similar to the following appear:

   ```
   ==Checking Directory Server Setup from comm_dssetup
   Domain Suffix: o=isp
   User/Group Suffix: o=isp
   Mail List User Suffix: o=mlusers
   Schema Type: 2
   ```

10. Type a mail address for postmaster notices.
    Select an address that your administrator actively monitors. For example, `pma@siroe.com` for a postmaster on the `siroe` domain. This address cannot begin with "Postmaster."

    > **Note**
    > The user of the email address is not automatically created (although the default "admin" user is automatically created). Therefore, you need create it later by using a provisioning tool.

11. Type the IP addresses of hosts that are permitted to relay mail without authentication.
    You can use the `$(IP-pattern/significant-prefix-bits)` syntax. This information creates the appropriate mapping entries. It is important that you modify your configuration to match the needs of your site. Specifically, your Messaging Server should recognize its own internal systems and subnets from which SMTP relaying should always be accepted. If you do not update this configuration, you might encounter problems when testing your MTA configuration. For more information, see Configuring SMTP Relay Blocking.
12. Type the password for administrator accounts.
    Type an initial password to be used for service administrator, server, user/group administrator, end user administrator privileges as well as PAB administrator and SSL passwords.
    After creating the initial runtime configuration, you might change this password for individual administrator accounts. For more information, see the topic on modifying your passwords in *Messaging Server Unified Configuration System Administrator's Guide*.
13. Verify the password for administration.
    Retype the administration password.
14. The program displays the changes that it makes as well post-configuration changes that you might want to make.

To Start Messaging Server

- To start Messaging Server, use the following command:

  ```
  cd <msg-svr-base>/bin
  ./start-msg
  ```

To Perform a Silent Installation

The Messaging Server initial runtime configuration program automatically creates a silent installation *state* file (called `saveState`) that can be used to quickly configure additional Messaging Server instances in your deployment where the Messaging Server packages have been installed. All of your responses to the configuration prompts are recorded in that file.

By running the silent installation, you instruct the `configure` program to read the silent installation state file. The `configure` program uses the responses in this file rather than ask the same installation questions again for subsequent initial runtime configurations of Messaging Server. When you use the state file in a new installation, you are not asked any questions. Instead, all of the state file responses are automatically applied as the new installation parameters.

The silent installation `saveState` file is stored in the *msg-svr-base*`/data/setup/` directory.

To use the silent installation file to configure another Messaging Server instance on another machine in the deployment, follow these steps:

1. Copy the `saveState` file to a temporary area on the machine where you are performing the new installation.
2. Review and edit the `saveState` file as necessary.
   The `saveState` file contains *parameter* = *value* pairs. Change parameters and values as needed. For example, the default email domain for the new installation might be different than the default email domain recorded in the `saveState` file. Remember that the parameters listed are automatically applied to this installation. Almost always, you need to change the host name (`Fqdn.TextField`). The `UGDIR_BINDPW` and `admin.password` fields are obfuscated but still need to be kept private.
3. Run the following command to configure other machines with the silent installation file:

   ```
   cd <msg-svr-base>/bin
   ./configure -state <statefile>
   ```

   where *statefile* is file name of the `saveState` file, including the full path to the file. (See Step 1 of this section).

   > **Note**
   > After running the silent installation program, a new state file is created from the silent installation in the *msg-svr-base*`/data/setup/` directory.

Configuring Messaging Server Against a Directory Server Replica

The following conditions might prevent you from configuring Messaging Server against a Directory Server host:

- You do not have Directory Server credentials.
- Messaging Server cannot communicate directly with the Directory Server master.

To Configure Messaging Server Against a Directory Server Replica

This task describes how to configure your deployment to be able to run Messaging Server against a Directory Server replica. You need to update the Directory Server master, which then feeds the replica with the necessary changes. You cannot update the Directory Server replica directly because the master Directory Server overwrites it.

1. Run the Messaging `configure` program using the replicated Directory Server credentials as described in Creating the Initial Messaging Server Runtime Configuration.
   Use the `--ldif` option to produce the *msg-svr-base*`/data/install/configure.ldif` file that is needed to allow proper privileges to the Directory Server.
2. Move the `configure.ldif` file to the Directory Server master.
3. Run the `ldapmodify` command on the `configure.ldif` file.
   Once the changes are replicated to the Directory Server replica, it is now configured to work with your Messaging Server.

Installing Messaging Server Provisioning Tools

To learn more about the schema and provisioning options for Messaging Server, see the topic on understanding schema and provisioning options in *Unified Communications Suite Deployment Planning Guide*.

Configuring SMTP Relay Blocking

Starting with Messaging Server 7 Update 5, the `configure` program prompts you to enter host IP addresses that are allowed as SMTP relay hosts. The `configure` program uses this information to construct the appropriate mapping entries.

By default, Messaging Server is configured to block attempted SMTP relays. That is, Messaging Server rejects attempted message submissions to external addresses from unauthenticated external sources (external systems are any other system than the host on which the server itself resides). This default configuration is quite aggressive in blocking SMTP relaying in that it considers all other systems to be external systems.

IMAP and POP clients that attempt to submit messages by using Messaging Server system's SMTP server destined for external addresses, and which do not authenticate using SMTP AUTH (SASL), find their submission attempts rejected. Which systems and subnets are recognized as internal is typically controlled by the `INTERNAL_IP` mapping table. In Unified Configuration, this mapping table is part of the overall configuration, and is viewed or edited by using the `msconfig` command. In legacy configuration, this mapping table is found in the *msg-svr-base*`/config/mappings` file.

For instance, on a Messaging Server system whose IP address is `192.45.67.89`, the default `INTERNAL_IP` mapping table would appear as follows:

```
INTERNAL_IP
$(192.45.67.89/32) $Y
127.0.0.1 $Y
* $N
```

The initial entry, using the `$(IP-pattern/significant-prefix-bits)` syntax, is specifying that any IP address that matches the full 32 bits of `192.45.67.89` should match and be considered internal. The second entry recognizes the loopback IP address `127.0.0.1` as internal. The final entry specifies that all other IP addresses should not be considered internal.

You can add additional entries by specifying additional IP addresses or subnets before the final `$N` entry. These entries must specify an IP address or subnet (using the `$(.../...)` syntax to specify a subnet) on the left side and $Y on the right side. Or you can modify the existing `$(.../...)` entry to accept a more general subnet.

For instance, if this same sample site has a class C network, that is, it owns all of the `192.45.67.0` subnet, then the site would want to modify the initial entry so that the mapping table appears as follows:

```
INTERNAL_IP
$(192.45.67.0/24) $Y
127.0.0.1 $Y
* $N
```

Or if the site owns only those IP addresses in the range `192.45.67.80-192.45.67.99`, then the site would want to use:

```
INTERNAL_IP
! Match IP addresses in the range 192.45.67.80-192.45.67.95
$(192.45.67.80/28) $Y
! Match IP addresses in the range 192.45.67.96-192.45.67.99
$(192.45.67.96/30) $Y
127.0.0.1 $Y
* $N
```

The *msg-svr-base*`/bin/imsimta test -match` utility can be useful for checking whether an IP address matches a particular `$(.../...)` test condition. The `imsimta test -mapping` utility can be more generally useful in checking that your `INTERNAL_IP` mapping table returns the desired results for various IP address inputs.

After modifying your `INTERNAL_IP` mapping table, be sure to issue the *msg-svr-base*/`bin/imsimta cnbuild` (if you are using a compiled configuration) and the *msg-svr-base*`/bin/imsimta restart` utilities so that the changes take effect.

Further information on the mapping file and general mapping table format, as well as information on `imsimta` command line utilities, can be found in *Message Server Administration Reference*. In addition, information on the `INTERNAL_IP` mapping table can be found in *Messaging Server System Administrator's Guide*.

Enabling Startup After a Reboot

You can enable Messaging Server startup after system reboots by using the bootup script. On Linux, this script is *msg-svr-base*`/data/install/Sun_MsgSvr`. For Solaris OS 10, you should use the Service Management Framework. That is, by default, Messaging Server is not restarted after a system reboot unless you run this script. In addition, this script can also start up your MMP, if enabled.

To Enable Messaging Server After a Reboot

1. Copy the *msg-svr-base*`/data/install/Sun_MsgSvr` script into the `/etc/init.d` directory.
2. Change the following ownerships and access modes of the `Sun_MsgSvr` script:

Ownership (chown(1M))	Group Ownership (chgrp(1M))	Access Mode (chmod(1M))
`root` (superuser)	`sys`	0744

3. Change directories to the `/etc/rc2.d` directory and create the following link:

```
ln /etc/init.d/Sun_MsgSvr S92Sun_MsgSvr
```

4. Change directories to the `/etc/rc0.d` directory and create the following link:

```
ln /etc/init.d/Sun_MsgSvr K08Sun_MsgSvr
```

Performance and Tuning

Refer to the topic on performance tuning considerations for a Messaging Server architecture in *Messaging Server System Administrator's Guide*.

Post-Installation Directory Layout

After installing Messaging Server, its directories and files are arranged in the organization described in the following table. The table shows only those directories and files of most interest for typical server administration tasks.

Post-Installation Directories and Files

Directory	Default Location and Description
Messaging Server Base msg-svr-base	/opt/sun/comms/messaging/ or /opt/sun/comms/messaging64/ (default location) The directory on the Messaging Server machine dedicated to holding the server program, configuration, maintenance, and information files. To configure more than one Messaging Server base directory per machine, see the topic on using the ALTROOT command-line argument in *Unified Communications Suite Installation and Configuration Guide*.
Configuration config	*msg-svr-base*/config/ Contains all of the Messaging Server configuration files, such as config.xml for Unified Configuration, or the imta.cnf and the msg.conf files, for legacy configuration. This directory is symbolically linked to the config subdirectory of the data and configuration directory (default: /var/opt/sun/comms/messaging64/) that you specified in the initial runtime configuration.
Log log	*msg-svr-base*/log/ A convenience symbolic link to *msg-svr-base*/data/log, which contains the Messaging Server log files like the mail.log_current file.
Data data	*msg-svr-base*/data/ Contains databases, configuration, log files, site-programs, queues, store and message files. The data directory includes the config and log directories. This directory is by default symbolically linked (on UNIX platforms) to the data and configuration directory (default: /var/opt/sun/comms/messaging64) that you specified in the initial runtime configuration.
System Administrator Programs bin	*msg-svr-base*/bin/ Contains the Messaging Server system administrator executable programs and scripts such as imsimta, configutil, stop-msg, start-msg, and uninstaller.
Library lib	*msg-svr-base*/lib/ Contains shared libraries, private executable programs and scripts, daemons, and non-customizable content data files. For example: imapd and qm_maint.hlp.
SDK Include Files include	*msg-svr-base*/include/ Contains Messaging header files for Software Development Kits (SDK).
Examples examples	*msg-svr-base*/examples/ Contains the examples for various SDKs.
Installation Data install	*msg-svr-base*/data/install/ and *msg-svr-base*/data/setup/ Contains installation-related data files such as installation log files, silent installation files, factory default configuration files, and the initial runtime configuration log files.

Post-Installation Port Numbers

In the installation and initial runtime configuration programs, port numbers are chosen for various services. These port numbers can range from 1 to 65535. Select numbers that do not conflict with port numbers used by enabled system services or other third-party software. The authoritative list of registered port numbers is available at http://www.iana.org. The `/etc/services` also lists a subset of these numbers.

The following tables list the port numbers that are designated after installation.

Port Numbers Designated During Installation: Unified Configuration

Service	Port	Unified Configuration Option to Change Port	Unified Configuration Option to Enable/Disable Service
Message Store			store.enable (1)
IMAP Server	143	imap.port	imap.enable (1)
POP Server	110	pop.port	pop.enable (1)
IMAPS Server	993	imap.sslport	imap.enablesslport (0)
POPS Server	995	pop.sslport	pop.enablesslport (0)
LMTP Server	225	dispatcher.service: LMTP.tcp_ports	dispatcher.service:LMTP.enable
MTA			
SMTP Relay	25	dispatcher.service: SMTP.tcp_ports	dispatcher.service: SMTP.enable
SMTP Submit	587	dispatcher.service: SMTP_SUBMIT.tcp_ports	dispatcher.service: SMTP_SUBMIT. enable
SMTPS Submit	465	dispatcher.service: SMTP_SUBMIT.tcp_ports	dispatcher.service: SMTPS_SUBMIT. enable
http mail proxy	8990	http.port	http.enable (1)
https mail proxy	8991	http.sslport	http.enablesslport (0)
MMP			mmp.enable (0)
IMAP Proxy	143	imapproxy.tcp_listen: imapproxy1.tcp_ports	
POP Proxy	110	popproxy.tcp_listen: popproxy1.tcp_ports	
Submit Proxy	587	submitproxy.tcp_listen: popproxy1.tcp_ports	
IMAPS Proxy	993	proxyimapssl	
POPS Proxy	995	popproxy.tcp_listen: ssl_ports	
Submits Proxy	465	submitproxy.tcplisten: ssl_ports	
Internal Servers			
watcher	49994	watcher.port	watcher.enable (1)
job_controller	27442	job_controller.tcp_ports	mta.enable (1)
ENS	7997	ens.port	ens.enable (0)

Port Numbers Designated During Installation: Legacy Configuration

Service	Port	Legacy Configuration Parameter to Change Port	Legacy Configuration Parameter to Enable/Disable Service
Message Store			local.store.enable (1)
IMAP Server	143	service.imap.port	service.imap.enable (1)
POP Server	110	service.pop.port	service.pop.enable (1)
IMAPS Server	993	service.imap.sslport	service.imap.enablesslport (0)
POPS Server	995	service.pop.sslport	service.pop.enablesslport (0)
LMTP Server	225	dispatcher.cnf	dispatcher.cnf (disabled)
MTA			local.imta.enable (1)
SMTP Relay	25	dispatcher.cnf	dispatcher.cnf (enabled)
SMTP Submit	587	dispatcher.cnf	dispatcher.cnf (enabled)
SMTPS Submit	465	dispatcher.cnf	dispatcher.cnf (disabled)
http mail proxy	8990	service.http.port	local.http.enable (1)
https mail proxy	8991	service.http.sslport	service.http.enablesslport (0)
MMP			local.mmp.enable (0)
IMAP Proxy	143	Aservice.cfg	Aservice.cfg (0)
POP Proxy	110	Aservice.cfg	Aservice.cfg (0)
Submit Proxy	587	Aservice.cfg	Aservice.cfg (0)
IMAPS Proxy	993	Aservice.cfg and ImapProxyAService.cfg	Aservice.cfg and ImapProxyAService.cfg (disabled)
POPS Proxy	995	Aservice.cfg and PopProxyAService.cfg	Aservice.cfg and PopProxyAService.cfg (disabled)
Submits Proxy	465	Aservice.cfg and SmtpProxyAService.cfg	Aservice.cfg and SmtpProxyAService.cfg (0)
Internal Servers			
watcher	49994	local.watcher.port	local.watcher.enable (1)
job_controller	27442	job_controller.cnf	local.imta.enable (1)
ENS	7997	local.ens.port	local.ens.enable (0)

JMQ Notification

Messaging Server can use Oracle GlassFish Message Queue, a standards-based messaging service, to send event notifications. Message Queue is provided as a shared component when you install Messaging Server or other Communications Suite products.

For More Information
See the overview of JMQ notification in *Messaging Server System Administrator's Guide* for more information on integrating JMQ and Messaging Server.

Configuring Certificate Based Authentication

Messaging Server supports client certificate authentication. Support for dynamic CRL updates was introduced in Messaging Server 7 Update 4 and was "back-ported" to Messaging Server 7 Update 3.

For More Information
See *Unified Communications Suite Certificate Authentication Guide*.

Chapter 7. Messaging Server 8.0 Release Notes

Oracle Communications Messaging Server 8.0 Release Notes

These Release Notes contain important information available at the time of the general release of Oracle Communications Messaging Server 8.0.

Topics:

- About Messaging Server 8.0
- New Features in This Release of Messaging Server
- Deprecated and Removed Features for Messaging Server
- Requirements for Messaging Server 8.0
- Messaging Server Installation Notes
- Problems Fixed in This Release of Messaging Server
- Known Problems in Messaging Server
- Redistributable Files for Messaging Server

About Messaging Server 8.0

Messaging Server is a high-performance, highly secure messaging platform that can scale from thousands to millions of users. It provides extensive security features that help ensure the integrity of communications through user authentication, session encryption, and the appropriate content filtering to reduce spam and viruses. With Messaging Server, enterprises and service providers can provide secure, reliable messaging services for entire communities of employees, partners, and customers.

Messaging Server provides a powerful and flexible solution to the email needs of enterprises and messaging hosts of all sizes by using open Internet standards.

New Features in This Release of Messaging Server

See New Features in Messaging Server 8.0.

Deprecated and Removed Features for Messaging Server

Support for the following features may be eliminated in a future release, may be already removed in this release, or removed in a previous release:

- Removal of MoveUser Command
- Removal of IMAP XSENDER Command
- Oracle GlassFish Message Queue is Deprecated
- Removal of the JMQ Default Password
- Support for Accessing Berkeley DB Databases has been Removed from the MTA.
- MMP Legacy Configuration Support is Deprecated
- Removal of MMP Legacy Log Format
- Deprecation of msgcert
- Change of local.sslv3enable default
- Deprecation of MoveUser and msgssh Commands (formerly msgadm)
- Red Hat Linux 32-bit Version and Red Hat Linux 4
- Deprecation of the readership Command
- MTA BDB Databases

- SIMS 4.0 IMTA SDK
- Oracle GlassFish Message Queue
- Sparse Zones
- Deprecation of Enabling POP Before SMTP
- Deprecation of imexpire -s Feature
- native, unix and file mailDeliveryOption Settings Deprecated
- Deprecation of Support for TLS Features Described as "must not" or "should not" in TLS Best Practices
- Messaging Multiplexor's (MMP) default:SSLSecModFile Option Removed
- shim64 Code Removed from Messaging Server
- The imsimta cache -rebuild Command Removed

Removal of MoveUser Command

The `MoveUser` command has been removed from this release of Messaging Server. The `MoveUser` command is inferior to the `rehostuser` command for moving users within a deployment, and is inferior to `imsbackup` and `imsrestore` for moving users from old to new product installations. Third-party `imapcopy` utilities are available for moving users between IMAP servers from different vendors.

Removal of IMAP XSENDER Command

The `XSENDER` command has been removed the from the IMAP server. If MMP `imapproxy.capability` is explicitly configured, please make sure XSENDER is not included in the value.

Oracle GlassFish Message Queue is Deprecated

The Oracle Glassfish MQ C SDK (also known as OpenMQ and JMQ) and JMQ JMS provider are not recommended. They have been deprecated and their support may be removed in a later release. Instead, use Java JMS (presently with the Oracle Glassfish MQ provider) and the ENS C API that Oracle Communications Mobile Synchronization Gateway uses and provides. Note that we do not support use of JMQ with anything running in web containers other than Glassfish.

Removal of the JMQ Default Password

The JMQ default password has been removed in this release of Messaging Server. The JMQ notification plugins that used to work using the default password will no longer work until the password is explicitly set in the configuration.

Support for Accessing Berkeley DB Databases has been Removed from the MTA.

Note that the various ancillary utilities, in particular `imsimta dumpdb`, have not been removed so customers may continue to access any data they may have stored in existing MTA databases.

Additionally, facilities have been provided to use the `memcache` protocol as an alternative for direct use of Berkeley DB. Note that `memcachedb` provides `memcache` protocol access to Berkeley DB; it could be used to continue storing MTA information in Berkeley DB, except with the advantage that multiple systems could share the same database.

The following MTA options control the use of the `memcache` protocol with various MTA databases:

`GENERAL_DATABASE_URL` General database
`REVERSE_DATABASE_URL` Reverse database
`FORWARD_DATABASE_URL` Forward database

`DOMAIN_DATABASE_URL` Domain database
`ALIAS_DATABASE_URL` Alias database
`SSR_DATABASE_URL` Server side sieve rules database

Each of these options can be used to specify a `memcache` URL of the form:

```
memcache://host:port/key-prefix
```

If the host isn't specified as part of the URL it defaults to the value of the `MEMCACHE_HOST` MTA option. It is an error for `MEMCACHE_HOST` not to be set in this case.

If the port isn't specified it defaults to the value of the `MEMCACHE_PORT` MTA option; if that option in turn isn't specified the default is 11211, the usual port for `memcache` servers.

Key-prefix, if specified, is prepended to the keys the duplicate extension sends to the `memcache` server.

The `imsimta crdb` utility has been extended to support loading data via the `memcache` protocol. This option is activated simply by specifying a `memcache:` URL instead of a destination file. A `-timeout` qualifier can be used to specify the timeout value to attach to the entries that are created.

The `imsimta test -db` utility can be used to test this new functionality in various ways. For example, assuming the `GENERAL_DATABASE_URL` MTA option is set to an appropriate `memcache: URL`, the following commands will test the ability to add, retrieve, and delete database entries.

```
% imsimta test -db -database=general
1000 entries processed, 1000 failures
% imsimta test -db -database=general -add
1000 entries processed, 0 failures
% imsimta test -db -database=general
1000 entries processed, 0 failures
% imsimta test -db -database=general -delete
1000 entries processed, 0 failures
% imsimta test -db -database=general
1000 entries processed, 1000 failures
```

This test uses an ascending sequence of entry values. Adding `-random=`*key* will use random hash values instead. `-repetitions` can be used to specify the number of test entries; the default is 1000.

MMP Legacy Configuration Support is Deprecated

MMP support for legacy configuration is deprecated in this release and may be removed in a later release.

Removal of MMP Legacy Log Format

The MMP legacy log format has been removed from this release. It was enabled by the `use_nslog` option, which the MMP now ignores.

Deprecation of msgcert

This command has been removed in Messaging Server 7 Update 5. The `msgcert` command's key generation and certificate request capabilities are obsolete due to recent weakness in MD5 and the NIST 2010 guidelines for SSL security strength. Use `certutil` with appropriate options (`-Z SHA1 -g 2048`) or other third-party certificate generation tools to create certificates and certificate requests with up-to-date security strength. See *Unified Communications Suite Certificate Authentication Guide* for more information on using `certutil`.

Change of local.sslv3enable default

The is `local.sslv3enable` parameter determines whether legacy support for the SSLv3 protocol (as opposed to the modern TLS protocol) is enabled. The security community considers SSLv3 deprecated and thus the default for this option has been changed to `0`.

Deprecation of MoveUser and msgssh Commands (formerly msgadm)

These two ancillary utilities are now considered deprecated. There are no plans to enhance these utilities and they may be removed in a future release.

The `MoveUser` command is inferior to the `rehostuser` command for moving users within a deployment, and is inferior to `imsbackup` and `imsrestore` for moving users from old to new product installations. Third-party `imapcopy` utilities are available for moving users between IMAP servers from different vendors.

A regular `ssh` session with appropriately configured `RBAC` (Solaris OS) or `sudo` (Red Hat Linux) provides enhanced security when compared to the `msgssh` command. In addition, an `ssh` session provides a more flexible framework for remote administration, because it supports administration of co-located products in addition to just Messaging Server.

Red Hat Linux 32-bit Version and Red Hat Linux 4

Support for the 32-bit Red Hat Linux version of Messaging Server and support for the Red Hat Linux 4 platform is deprecated and may be removed in a future release.

Deprecation of the readership Command

Support for the `readership` command is deprecated and may be removed in a later release.

MTA BDB Databases

MTA access to database files and the imsimta tools to manipulate MTA database files have been deprecated since the Messaging Server 6 release, and may be removed in a future release. MTA text databases continue to be supported.

SIMS 4.0 IMTA SDK

The SIMS 4.0 IMTA SDK has been deprecated since iPlanet Messaging Server 5 was released and may be removed in a future release. The current MTA SDK remains supported.

Oracle GlassFish Message Queue

Do not use the Oracle Glassfish MQ C SDK (also known as OpenMQ and JMQ), as it is deprecated. Oracle reserves the right to change the underlying protocol and the JMS provider used to provide Java JMS support for notifications in a future release. In the event this happens, the JMQ C SDK ceases to interoperate. Use of Java JMS (presently with the Oracle Glassfish MQ provider) and the ENS C API is supported.

Sparse Zones

Sparse zone support is deprecated and may be removed in a future release.

Deprecation of Enabling POP Before SMTP

SMTP Authentication, or SMTP Auth (RFC 2554) is the preferred method of providing SMTP relay server security. SMTP Auth allows only authenticated users to send mail through the MTA. The MMP has a legacy POP before SMTP feature. This feature is now deprecated and may be removed in a future release.

Deprecation of imexpire -s Feature

This feature is deprecated and may be removed in a later release.

native, unix and file mailDeliveryOption Settings Deprecated

The `native`, `unix` and `file` `mailDeliveryOption` settings are deprecated and may be removed in a later release.

If you actively depend on these features please contact Oracle support.

Beginning with Messaging Server 7 Update 5, the initial unified configuration will no longer include a channel block and channel class for the native channel. The `native` and `file` delivery options will not work by default. There is no expected impact to customers using an existing configuration that is upgraded at this time.

Deprecation of Support for TLS Features Described as "must not" or "should not" in TLS Best Practices

Support is deprecated for all TLS features mentioned as "must not" or "should not" in http://tools.ietf.org/html/draft-ietf-uta-tls-bcp and may be removed in a later release.

Messaging Multiplexor's (MMP) default:SSLSecModFile Option Removed

The Messaging Multiplexor's (MMP) `default:SSLSecModFile` option has been removed and is no longer honored.

The Messaging Multiplexor uses the NSS shared DB feature by default.

shim64 Code Removed from Messaging Server

We have removed shim64 code from the product. 32-bit spamfilter plugins will no longer work. Brightmail customers may request a 64-bit SDK that will work natively with Messaging Server.

The imsimta cache -rebuild Command Removed

The `imsimta cache -rebuild` command is no longer useful and has been removed from the product and the `imsimta cache` documentation. Any remaining scripts that employ the command should replace it with the sequence:

```
stop-msg job_controller
start-msg job_controller
```

Requirements for Messaging Server 8.0

Supported Operating Systems

The following table lists the operating systems that support Messaging Server.

Operating System	CPU
Oracle Solaris 10 and 11	SPARC, X64
Oracle Linux and Red Hat Enterprise Linux 6 64-bit	X64

Required Software

The following table lists the software required for installing and running Messaging Server.

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.
Directory Server Setup Script (`comm_dssetup.pl`)	You must use the version that is bundled with the Messaging Server installer.	To prepare the LDAP directory for Messaging Server.

> **Note**
> For information about upgrading to Messaging Server 8.0 from a previous version of Messaging Server, see Messaging Server Installation Notes.

Messaging Server Installation Notes

These installation notes pertain to the Messaging Server 8.0 release. This section contains the following subsections:

- Installation Overview for Messaging Server
- Upgrade Instructions for Messaging Server

Installation Overview for Messaging Server

Use the `commpkg` installer to install Messaging Server.

For installation instructions, see the Messaging Server 8.0 installation scenarios:

- Installation Scenario - Messaging Server 8.0 Message Store
- Installation Scenario - Messaging Server 8.0 Message Transfer Agent
- Installation Scenario - Messaging Server 8.0 Messaging Multiplexor
- Installation Scenario - Messaging Server 8.0 Webmail Server

After installation is complete, you must configure Messaging Server by:

1. Running the Directory Server Preparation Tool, `comm_dssetup.pl`
2. Running the Messaging Server configuration program

For configuration instructions, see Messaging Server 8.0 Initial Configuration.

Upgrade Instructions for Messaging Server

If you are upgrading to Messaging Server 8.0 from an earlier release, follow the upgrade instructions in Messaging Server 8.0 Upgrade.

Problems Fixed in This Release of Messaging Server

Problems Fixed in Messaging Server 8.0

The following table lists problems fixed in Messaging Server 8.0.

Problems Fixed

Service Request (SR) Number	BugDB Number	Description
3-10014737481	20236257	Need a source-channel-specific way to make the 5yz a 4yz error
NA	18740251	Add the `destination passthrough` to the tcp_local channel during initial configuration
3-8390376431	18390240	Message Tracing does not record APPEND operations
3-8602857581	18322860	`imcheck -s` to show db free space
3-8544479561	18240417	Initial configure should warn or not overwrite critical LDAP attributes
3-10057644961, 3-8108546381, 3-8192540841	17821765	`watcher` kills process to be restarted and then `msstart` declines to restart
3-7852088911	17622030	`msstart` should not try others, should fail more quickly when `stored` fails
3-7656654731	17312275	`ims_svc_start` should not log failure at info level
3-7436182911	17045338	"Message contains invalid header" error on APPEND
3-7147737511	16963853	`rehostuser` exit codes are not from `sysexits.h`
3-7223715641	16821861	General chattyness of ENS logging
3-7205696131	16808116	MS parameter for connection timeout on `libmilter.so`
3-6851086021	16402816	Need the Messaging Server LDAP client to support LDAP STARTTLS authentication
3-6839167541	16383228	`tcp_lmtp_server` core on shutdown in `mqueue_close`
3-7841143071, 3-6514695211	15947899	IMAP APPEND needs to be able reject large messages like MTA options
3-8269502341, 3-5690858563	14064118	Startup/reconnect bottleneck on `setaccess`
3-8390376431	13866116	Need counter of IMAP APPEND operations

Red Hat Enterprise Linux 5 SNMP

Red Hat Enterprise Linux 5 only supports use of Simple Network Management Protocol (SNMP) by 64-bit products. If you want to use the 32-bit version of Messaging Server, SNMP support is only available on Red Hat Enterprise Linux 4. The 64-bit version of Messaging Server does include SNMP support on Red Hat Enterprise Linux 5.

Known Problems in Messaging Server

Known Problems in Messaging Server 8.0

This section describes known problems in Messaging Server 8.0.

Messaging Server Fails to Start After Upgrading From Messaging Server 7 Update 4 Patch 27 to Messaging Server 8.0 in Sun Cluster

SR number: NA
Bug number: 20810772

In a highly available deployment using Sun Cluster, after upgrading from Messaging Server 7 Update 4 Patch 27 to Messaging Server 8.0, Messaging Server fails to start and the following message is displayed:

```
scswitch: (C969069) Request failed because resource group
<messaging_server_resource_group> is in
ERROR_STOP_FAILED state and requires operator attention
```

Workaround:

Perform the following on the Sun Cluster active node:

- After completing the upgrade to Messaging Server 8.0, and before executing the `scswitch` command (to switch the resource group back to the active node), run the following commands:

```

start-msg watcher
configutil -o local.store.notifyplugin.ms-internal.ensport -v 7997
stop-msg ha
```

Message Store Reads ldap_host_alias_list From the MTA Option Section

SR number: NA
Bug number: 20764412

The message store uses an MTA option (`ldap_host_alias_list`) to determine the set of local host aliases. However, the message store currently reads this option from the MTA option section, not the store or base section.

When used in Unified Configuration mode, the `ldap_host_alias_list` option could cause an issue where the MTA version of the option is no longer seen by the message store.

MS_SCHA Agent Binary Files Should Be Owned by Root User

SR number: NA
Bug number: 17470656

During HA installation, Messaging Server resource creation failed because agent binary files were owned by `bin:bin`.

Workaround: Go to the MS_SCHA agent `bin` directory and change ownership of all files to `root:root`.

For example:

```
# cd /opt/sun/comms/msg_scha/bin
# chown root:root *
```

Redistributable Files for Messaging Server

The following redistributable files are provided with Messaging Server:

- You can copy and use (but not modify) the following header files solely to create and distribute programs to interface with Messaging Server APIs, to compile customer written code using the documented API to interoperate or integrate with Messaging Server, and only as expressly provided in the Messaging Server documentation:
  - *msg-svr-base*`/examples/tpauthsdk/authserv.h`
  - All files in the *msg-svr-base*`/include` directory (default location)

- The following files are provided solely as reference for writing programs that use the documented API to integrate with Messaging Server:
  - *msg-svr-base*`/examples/tpauthsdk/`
  - *msg-svr-base*`/examples/mtasdk/`

Chapter 8. Messaging Server 8.0 Sun Cluster HA Agent Initial Configuration

Oracle Communications Messaging Server 8.0 Sun Cluster HA Agent Initial Configuration

After installing the Messaging Server Sun Cluster HA Agent software, you need to perform an initial configuration by running the following command:

```
<msg-scha-base>/bin/init-config
```

This command registers the HA agent with the Sun Cluster HA software. You must have the Sun Cluster HA software installed prior to issuing this command.

For Messaging Server 7 Update 2 Only
After configuring Messaging Server 7 Update 2 for HA or upgrading to Messaging Server 7 Update 2 in HA, if you are using a compiled configuration, you must recompile the configuration by issuing the command:

```
imsimta cnbuild
```

Otherwise, the Messaging Server fails to start in the HA environment.

Chapter 9. Messaging Server 8.0 Upgrade

Oracle Communications Messaging Server 8.0 Upgrade

Caution
Once you upgrade to Messaging Server 7.0.5 or greater, including Messaging Server 8.0, from a version prior to Messaging Server 7.0.5, you cannot downgrade by "backing out" the upgrade. This is because of database incompatibilities with prior versions starting in Messaging Server 7.0.5. For instructions on returning to a previous version after upgrading to Messaging Server 8.0, see Downgrading From Messaging Server 8.0.

This information describes the three Messaging Server upgrade strategies and procedures to upgrade from Messaging Server 7.x to Messaging Server 8.0. It assumes that you have chosen a target deployment, and have developed an architectural design and deployment plan.

Topics:

- Messaging Server Upgrade Requirements
- New Upgrade Features in Messaging Server 8.0
- About Messaging Server Unified Configuration
- Upgrading Messaging Server Overview
- Messaging Server Upgrade Strategies
- Using the Side-by-Side Strategy to Upgrade Messaging Server
- Messaging Server 8.0 Side-By-Side Upgrade
- Using the In-Place Upgrade on Messaging Server
- Upgrading Messaging Server with Webmail Over IMAP Protocol

Note
If you are upgrading from Sun Java System Messaging Server 5.2, see the topic on coexistent upgrades from iPlanet Messaging Server 5.2 in *Unified Communications Suite 6 Update 1 Installation and Configuration Guide*.

Messaging Server Upgrade Requirements

The requirements for upgrading to Messaging Server 8.0 are:

- You must be running Messaging Server 7.x to upgrade to Messaging Server 8.0.
- You cannot upgrade from Messaging Server 5.x or 6.x directly to Messaging Server 8.0. You must first upgrade to Messaging Server 7.x, then upgrade to Messaging Server 8.0. Contact Oracle Consulting to upgrade directly from Messaging Server 5.x or 6.x to Messaging Server 8.0.
- Linux platforms: Messaging Server 8.0 only supports Oracle Linux/Red Hat Enterprise Linux 6.x.

Note
This document uses the side-by-side installation method to be consistent between Solaris and Linux platforms. In general, you should avoid using the alternate root method when upgrading Messaging Server, because Solaris now uses alternate root for its Live Upgrade feature.

New Upgrade Features in Messaging Server 8.0

The Messaging Server 8.0 upgrade includes the following changes and new features, which simplify the side-by-side upgrade method:

- Upgrade Does Not Touch Messaging Server Data or Configuration
- Improvements to the stored -r Command
- Solaris SRV4 Patches

Upgrade Does Not Touch Messaging Server Data or Configuration

Starting with version 8.0, Messaging Server package scripts and `preupgrade` and `postupgrade` scripts no longer alter the data and configuration in any way. In addition, the upgrade no longer automatically runs the `stop-msg` command when uninstalling.

For side-by-side migrations, this feature enables you to install two separate Messaging Server versions, such as 7.0.5 and 8.0, on the same host, that point to the same data and configuration, and activate a version by running that version's specific `start-msg` command. The Messaging Server data and configuration are "upgraded" when the `start-msg` script invokes the `updateCfgVersion` script after detecting that a new Messaging Server version is used for the first time.

Improvements to the stored -r Command

Starting with version 8.0, Messaging Server upgrade no longer runs the `stored -r` command prior to uninstalling the previous version's binaries.

Solaris SRV4 Patches

Starting with version 8.0, Messaging Server SVR4 style patches are no longer available on Solaris. Instead, you use Automated Release Update (ARU) patches. ARU patches treat each Messaging Server 8.0 and subsequent versions as a different package version. For example, Messaging Server 8.0 has a different package version than Messaging Server 8.0 patch 1. Because of this versioning, you can install two copies of the same version of Messaging Server on the same host. Thus, for upgrades, you no longer need to use the alternate root (ALTROOT) install method.

About Messaging Server Unified Configuration

Beginning with Messaging Server 7 Update 5, Messaging Server has the capability to create a Unified Configuration. Unified Configuration provides an improved, streamlined process to configure and administer Messaging Server. Unlike in legacy configurations (Messaging Server 7 Update 4 and prior releases), Unified Configuration uses validation to verify configuration accuracy, and employs a single tool to configure the entire Messaging Server configuration (with a few exceptions). Thus, moving your deployment to Unified Configuration simplifies administration and reduces configuration mistakes.

After upgrading to Messaging Server 7 Update 5 and later, you can decide to migrate your legacy configuration to Unified Configuration. It is not a requirement to use Unified Configuration with Messaging Server 7 Update 5 and later, however, Unified Configuration provides a number of benefits over legacy configuration. When you convert to Unified Configuration, Messaging Server saves your old legacy configuration in the *configroot/*`legacy-config` directory. If necessary, you can restore a saved legacy configuration at the time you converted, however, all changes made to your configuration after converting to Unified Configuration are lost. You can migrate to Unified Configuration after you have completed the upgrade. You are not required to migrate to Unified Configuration during the upgrade process.

To help you decide to migrate to Unified Configuration, see the overview of Messaging Server Unified Configuration in *Messaging Server Unified Configuration System Administrator's Guide*.

Upgrading Messaging Server Overview

A Messaging Server deployment can consist of multiple back-end message stores, multiple Webmail servers, front-end MMPs, and MTA relays. Like all upgrades, you proceed on a host-by-host basis. Upgrading a Messaging Server deployment includes the following high-level steps:

- Backing up the Messaging Server data
- Upgrading and running `comm_dssetup.pl` to the latest version before upgrading Messaging Server
  Messaging 8.0 requires you to apply at least `comm_dssetup.pl` *version 6.4.0.27.0* against Directory Server. The Messaging Server 8.0 media pack includes `comm_dssetup.pl` version 6.4.0.27.0.
- Defining your upgrade target and the required products and components for that target
- Reviewing your Messaging Server architecture and topology
  Although you might be satisfied with your current Messaging Server architecture and topology, upgrading can provide the opportunity to redesign your deployment for more optimal performance. Refer to *Unified Communications Suite Deployment Planning Guide* for more information.
- Selecting the upgrade sequence of individual Messaging Server hosts
  This includes upgrading components such as the message store servers, proxies, Webmail servers, and front-end relays.
- Choosing a Messaging Server upgrade strategy for each host
  Three Messaging Server upgrade strategies offer choices that strike a balance between system downtime, cost, simplicity, and risk. You choose a strategy for each host, and you can use different strategies on different hosts within a Messaging Server deployment.

  > **Note**
  > As of Communications Suite 7, Messaging Server 32-bit has been dropped on Oracle Solaris.

- Upgrading the Messaging Server software
  Use Messaging Server 8.0 or the current patch.
- Optional: Migrating to Unified Configuration
  Use the `configtoxml` command to migrate from legacy configuration to Unified Configuration. See the `configtoxml` command syntax in *Messaging Server Unified Configuration System Administrator's Guide* for more information.

Technical Features Supporting Messaging Server Upgrade

The following features support Messaging Server upgrade:

- You migrate mailboxes by using the `imsbackup` and `imsrestore` commands. See the topic on migrating mailboxes to a new system in *Messaging Server System Administrator's Guide*. These commands support moving mailboxes from old message store versions to new ones (including when the message store database format changes, for example, from Messaging Server 32-bit to Messaging Server 64-bit). These commands also support moving mailboxes from new message store versions to old ones for back-out purposes.
- In-place Upgrade supports changing the old mailbox format to the new format, but it does not support going from the new format back to the old. You **cannot** back out from new data format to old data format by using the in-place Upgrade Strategy. The conversion is done "on-the-fly" as mailboxes are accessed. In-place server upgrade is by done using the `commpkg upgrade` command.
- Migrating the Messaging Server configuration from the old system to the new system is done by using the `migrate-config` utility.
- Alternate root (ALTROOT) install is supported on Oracle Solaris. See the topic on using the ALTROOT command-line argument in *Unified Communications Suite Installation and Configuration Guide* for more information.

Note
In general, you should avoid using the alternate root method when upgrading Messaging Server, because Solaris now uses alternate root for its Live Upgrade feature.

Messaging Server Upgrade Strategies

Messaging Server supports the following three upgrade strategies for individual hosts. These strategies provide a balance between downtime, risk of extended downtime, complexity, and potential hardware costs.

- *In-place Upgrade.* The binaries of the old version are replaced with the binaries of the new version on the same host. That is, you use `commpkg upgrade`.
- *Side-by-side Upgrade on the same host.* The new software version is installed on the same host as the old version in a different directory. After you migrate the software configuration to the new version, you switch the deployment over to the new version.
- *Coexistent Upgrade.* You keep existing services online while you construct a new host on separate hardware.

The strategy chosen for any particular host might differ. For example, you might wish to use an in-place upgrade on your front-end servers (relays, MMPs, and webmail servers) but you might want to do a coexistent upgrade on your message stores.

Caution
There is a data format change in the message store in Messaging Server 8.0 (see the topic on upgrading the message store in *Messaging Server System Administrator's Guide*). Coexistent upgrade is recommended to facilitate backing out from an upgrade. See also Downgrading From Messaging Server 8.0 for additional information.

The strategy you chose also depends upon the version you currently have installed and whether you are using 32-bit or 64-bit Messaging Server product. Issues and compatibilities are described next.

Note
When upgrading/migrating between SPARC and x86 hardware, you need to use the Online/Coexistence strategy. Also, see the topic on migrating from x86 to SPARC in *Messaging Server System Administrator's Guide*.

The Coexistence Migration Strategy is the safest and most secure method of upgrading. It also has the lowest downtime of the three upgrade strategies. In the coexistence model, existing services remain online while you construct a new target host (or entire Messaging Server environment) on new hardware or in a Oracle Solaris whole root zone on the existing hardware. After the new host and environment are established, you can migrate a small number of friendly users to the new system to verify operations and administrative procedures. For a certain period both systems are accessible to user traffic. This is called a coexistence phase. Messaging access is not disrupted and proceeds invisibly to users. When all users are migrated to the new environment, you can decommission your legacy deployment. This phased approach ensures that the new system is fully prepared to handle production users before making the full migration.

> **Note**
> Read about coexistent upgrades From iPlanet Messaging Server 5.2 in *Unified Communications Suite 6 Update 1 Installation and Configuration Guide* for useful information on coexistent upgrades.

Advantages and Disadvantages of Coexistence Migration:

- Service downtimes are usually rare and short. There is less danger that they will be longer than the off-line windows imposed by service level agreements.
- Allows a gradual adoption of the new software so that you can gain confidence by trying it out with a small group of sympathetic users before migrating production users.
- The risk of upgrade failure is mitigated by the fact that your legacy system remains fully functioning throughout the upgrade process.
- Because the new system is built alongside a functional old one, you do not need to install or modify anything on the working legacy machines. This is an advantage as there is always a natural reluctance to modify or reconfigure a working legacy system in significant ways.
- Coexistence is the safest upgrade model and has the least amount of user downtime.
- Simpler back off procedure. Anytime you upgrade software, you need to make provisions for backing off from the new system to the old system in case of failure. Other upgrade models might require that you back up and turn off the old system, install, configure, and migrate to the new system. Only when you switch on the new system do you know if the upgrade succeeded. If it turns out, that it did not, then you might have to use your back off plan to put everything back into place. A coexistence migration is much simpler as a working legacy system is already in place.
- You must move user data, such as mailboxes, from one host to another, typically by using the `imsbackup` and `imsrestore` commands.
- Might require extra hardware to set up a parallel system. (This can be mitigated by upgrading legacy machines after they are no longer used.)

Specific Steps for Upgrading Messaging Server Using the Coexistence Model

1. Make sure that your hardware is installed as per the deployment plan created from Convergence deployment planning] and Communications Suite deployment planning.
2. Install new version of Messaging Server on new machine, by using the `commpkg install` command.
3. Configure Messaging Server.
   You must do so manually. Basically you must clone the old machine's configuration to this new machine.
4. If you are doing a coexistent migration on a message store, migrate user mailboxes (a few at a time) to the new machine. See the topic on migrating or moving mailboxes to a new system in *Messaging Server System Administrator's Guide*. Details on message store internals can be found in the topic on upgrading the message store in *Messaging Server System Administrator's Guide*.

Using the Side-by-Side Strategy to Upgrade Messaging Server

In this model, you install the new software version on the same machine as the old version. The basic steps are as follows:

1. Back up configuration and mailbox data just in case a back out is required.
   For the configuration data, simply back up the configuration directory. For mailbox data, use the `imsbackup` command.
2. Install Messaging Server 8.0 side-by-side on the same machine with your earlier version of Messaging Server by using the `commpkg install` command.
3. Create a symbolic link for a level of indirection that you will use to point to the active Messaging Server installation.

4. Stop the currently running Messaging Server.
5. Point the symbolic link to the Messaging Server 8.0 installation
6. Start Messaging Server 8.0.

Advantages and Disadvantages of Side-by-Side Messaging Server Migration

- Second best minimal downtime.
- Second best in backout.
- Does not require extra machines.
- Does require different directory location for fresh install. Any custom scripts that reference the install location must be modified.
- Does not involve moving the mailboxes. New version just "points" to the mailboxes and mailbox conversion to the new version is automatic and transparent.
- Back out is complicated and time consuming. See Downgrading From Messaging Server 8.0.
- The only advantage of side-by-side over in-place is that the binaries of the old version remain intact on the system so you do not have to reinstall and reconfigure in the case of a backout.

Messaging Server 8.0 Side-By-Side Upgrade

This example describes how to upgrade from Messaging Server 7.0.5.31.0 to Messaging Server 8.0 by using the side-by-side method.

Topics:

- Side-By-Side Migration Overview
- Side-By-Side Migration Example
- Handling Subsequent Upgrades

Side-By-Side Migration Overview

This example describes how to install both Messaging Server versions on the same host in separate directories, create a symbolic link to the active installation, then point the symbolic link at the single configuration and data location.

Note
Upgrading to Messaging Server 8.0 in a side-by-side installation works on both Solaris and Oracle Linux. This is not an alternate root installation as described in the topic on using the ALTROOT command-line argument in *Unified Communications Suite Installation and Configuration Guide*. Due to package version changes starting with Messaging Server 8.0, you can use the method described in this information rather than the alternate root method, to simplify the upgrade process.

This example uses the following directories:

- `/opt/sun/comms/messaging64`: Directory in which Messaging Server 7.0.5.31.0 is installed (default location)
- `/var/opt/sun/comms/messaging64`: Directory containing the Messaging Server 7.0.5.31.0 data and configuration (default location)
- `/opt/ucs1/messaging64`: Directory in which Messaging Server 8.0 is installed (non-default location)

Additionally, this example uses the following symbolic link:

- `/opt/ucs/msg`: Symbolic link to either `/opt/sun/comms/messaging64` or

```
/opt/ucs1/messaging64
```

Side-By-Side Migration Example

Topics:

- Backing Up Messaging Server
- Creating the Symbolic Link for the Active Messaging Server Installation
- Installing and Configuring Messaging Server 8.0
- Changing Over from Messaging Server 7.0.5.31.0 to Messaging Server 8.0
- Post Upgrade

Backing Up Messaging Server

Before performing the upgrade, back up the system. See the following documentation for more information:

- The topic on best practices for Messaging Server and ZFS in *Messaging Server System Administration Guide*.
- Downgrading From Messaging Server 8.0
- The topic on backing up and restoring the message store in *Messaging Server System Administration Guide*.

Creating the Symbolic Link for the Active Messaging Server Installation

This example assumes that you have already installed and configured Messaging Server 7.0.5.31.0 in the default directory (`/opt/sun/comms/messaging64`), and that the Messaging Server is currently running.

1. Create a symbolic link for a level of indirection that you will use to point to the active Messaging Server installation.

   ```
   mkdir -p /opt/ucs
   cd /opt/ucs
   ln -s /opt/sun/comms/messaging64 msg
   ```

2. Ensure that external programs or plugins that refer to the Messaging Server installation use this symbolic link. Also, if you use Solaris Management Facility (SMF), ensure that you configure XML settings that start and stop Messaging Server to use this symbolic link.

Installing and Configuring Messaging Server 8.0

1. Change to the directory in which you have extracted the Messaging Server 8.0 media pack ZIP file.
2. Install Messaging Server 8.0 into its own directory, `/opt/ucs1`, by using the following `commpkg install` command.

   ```
   commpkg install --comp=MS64 --installroot /opt/ucs1 --silent=NONE
   ```

3. Configure Messaging Server 8.0 to point to the existing (Messaging Server 7.0.5.31.0) data and configuration location.

```
cd /opt/ucs1/messaging64
bin/useconfig /var/opt/sun/comms/messaging64/config
```

Changing Over from Messaging Server 7.0.5.31.0 to Messaging Server 8.0

1. Stop the currently running Messaging Server 7.0.5.31.0 processes.

   ```
   /opt/ucs/msg/bin/stop-msg
   ```

   Note that this command actually uses the symbolic link to `/opt/sun/comms/messaging64`.

2. Change the symbolic link created previously to point to the Messaging Server 8.0 installation.

   ```
   cd /opt/ucs
   mv msg msg-old
   ln -s /opt/ucs1/messaging64 msg
   ```

3. Start the Messaging Server 8.0 processes.

   ```
   /opt/ucs/msg/bin/start-msg
   ```

   Note that this command actually uses the symbolic link to `/opt/ucs1/messaging64`.

Your deployment is now upgraded to Messaging Server 8.0.

Post Upgrade

After completing the upgrade, remove the symbolic links (data, config, and log) in the previous Messaging Server installation. This is not a requirement, but a recommendation to protect against inadvertently using them.

```
cd /opt/sun/comms/messaging64
rm data config log
```

Handling Subsequent Upgrades

On the next upgrade, now that the two locations are populated, you can simply upgrade the inactive location. Following the preceding example, Messaging Server 8.0, installed in `/opt/ucs1` is active, and Messaging Server 7.0.5.31.0, installed in `/opt/sun/comms` is inactive.

1. Change to the directory in which you have extracted the latest Messaging Server version media pack ZIP file.
2. If you are upgrading from a Messaging Server version prior to 8.0, for example, 7.0.5.31.0, you must remove the symbolic links to the configuration and data, otherwise the uninstall stops the messaging services.

```
cd /opt/sun/comms/messaging64
rm config data log
```

3. Upgrade the inactive Messaging Server installation.

   ```
   commpkg upgrade --comp=MS64
   ```

   The upgrade prompts you to select the version that you want to upgrade. Specify the inactive version.

4. Change the symbolic link created previously to point to the new Messaging Server installation.

   ```
   cd /opt/sun/comms/messaging64
   bin/useconfig /var/opt/sun/comms/messaging64/config
   ```

5. Stop the running Messaging Server processes.

   ```
   /opt/ucs/msg/bin/stop-msg
   ```

   Note that this command actually uses the symbolic link to `/opt/ucs1/messaging64`.

6. Change the symbolic link created previously to point to the new Messaging Server 8.0 installation. Depending on which installation you are upgrading, use one of the following `ln` commands.

   ```
   cd /opt/ucs
   rm msg
   ln -s /opt/sun/comms/messaging64 msg
   <or, depending on which installation is upgraded>
   ln -s /opt/ucs1/messaging64 msg
   ```

7. Start the Messaging services using the new, upgraded version.

   ```
   /opt/ucs/msg/bin/start-msg
   ```

8. You should remove the symbolic links in the inactive installation, otherwise you might inadvertently use the inactive installation.

Using the In-Place Upgrade on Messaging Server

In this method you simply replace the old server binaries with the new server binaries on the same machine by using the `commpkg upgrade` command. This command removes the old packages and installs the new ones. For details about this command, see the topic on commpkg upgrade usage in *Unified Communications Suite Installation and Configuration Guide*.

Advantages and Disadvantages of In-place Messaging Server Upgrade

- Simplest. One command installs the old packages and removes the new packages. This command migrates and upgrades configuration.
- Requires least amount of extra disk space.
- Messaging Server stays in the same disk location (no tweaking of custom scripts).
- Has the most downtime.
- Back out is complicated and time consuming. See Downgrading From Messaging Server 8.0.
- This method is probably best for evaluators/testers/developers.
- Useful for upgrading Messaging Servers configured without the message store, for example, front-end relays and webmail servers.

Specific Steps for Using In-Place Upgrade on Messaging Server

- Run `commpkg upgrade` and select Messaging Server.
    - Stops the servers.
    - Removes the old version.
    - Installs the new version.
    - Performs migration of configuration and mailbox data.

For information about using the `commpkg upgrade` command, see *Unified Communications Suite Installation and Configuration Guide*.

Upgrading Messaging Server with Webmail Over IMAP Protocol

Starting with Messaging Server 6.3, the webmail server (mshttpd) communicates with the message store by using IMAP. Thus, the HTTP service could be run on the front end and is no longer needed or enabled on the store. To support older MEM clients for coexistent migrations, enable `mshttpd` and configure it on the 7.x back-end message store systems.
For example:

```
configutil -o service.http.enable -v 1
```

The number of `mshttpd` processes (`service.http.numprocesses`) should not change on the 6.2 front end. However, you must set the number of processes on the new back end to 1.

Finally, you need to copy the webmail files in the *server-root*`/config/html` directory over from the 6.2 system to the 7.x back-end systems. Although the back-end `mshttpd` does not send JavaScript or HTML files to the front end, the contents and structure of that directory need to match the front end.

Chapter 10. Messaging Server 8.0 Upgrade in an HA Environment

Messaging Server 8.0 Upgrade in an HA Environment

Upgrading Messaging Server in a highly-available (HA) environment consists of upgrading the Messaging Server software then upgrading the Messaging Server Sun Cluster Agent.

Topics:

- Upgrading to Messaging Server 8.0 in an HA Environment
- Upgrading to the Messaging Server 7 Sun Cluster Agent (MS_SCHA)

Upgrading to Messaging Server 8.0 in an HA Environment

Upgrade strategies, each of which require different procedures, include the following:

- Coexistent upgrade: This is similar to a fresh HA installation. See the topic on configuring Messaging Server for high availability in *Messaging Server System Administrator's Guide* for more information
- Side-by-side upgrade
- In-place HA upgrade

To Do a Side-by-side Upgrade to Messaging Server 8.0 in an HA Environment

1. Go to the resource group online node.
   a. Disable Messaging server resource.

      ```
      # scswitch -n -j <msg_svr_resource>
      ```

   b. Upgrade Messaging Server by using the side-by-side strategy, see Side-by-Side Strategy to Upgrade Messaging Server. Perform this step only on the Messaging Server resource group online node. Do not start Messaging Server yet.
   c. Run the `ha_ip_config` command on the Messaging Server resource group online node.

      ```
      # <msg_svr_base>/sbin/ha_ip_config
      ```

      This command is needed only if the currently installed Messaging Server is prior to version 7.0.

2. Switch over to other node:

   ```
   # scswitch -z -g <msg_svr_resource_group> -h <node-name>
   ```

3. Run the `useconfig` command.
   This is needed if you are upgrading Messaging Server from 32-bit to 64-bit, to update the trusted

library path for 64-bit applications to include Messaging Server `/bin/crle -s -64` *`new_msg_svr_base`*`/lib'`).

```
# <msg_svr_base>/bin/useconfig <msg_svr_base>/config
```

4. Change `IMS_serverroot` path for Messaging Server resource if new Messaging Server base directory is different from old installation.

```
# scrgadm -cj <msg_svr_resource> -x IMS_serverroot=<new_msg_svr_base>
```

5. If Messaging Server Sun Cluster agent (`MS_SCHA`) is old (not from Communications Suite 6 or later), then it does not work with upgraded Messaging Server and you need to perform the `MS_SCHA` upgrade procedure.
6. Enable Messaging Server resource.

```
# scswitch -e -j <msg_svr_resource>
```

To Perform an In-place Upgrade to Messaging Server 8.0 in an HA Environment

An in-place upgrade is done by using the `commpkg upgrade` command.

1. Disable Messaging Server resource:

```
# scswitch -n -j <msg_svr_resource>
```

2. Run the `commpkg upgrade` command on all nodes of the cluster.
3. Run the `ha_ip_config` command on the Messaging Server resource group online node.

```
# <msg_svr_base>/sbin/ha_ip_config
```

This command is needed only if the currently installed Messaging Server is prior to version 7.0.

4. Enable Messaging Server resource:

```
# scswitch -e -j <msg_svr_resource>
```

Upgrading to the Messaging Server 7 Sun Cluster Agent (MS_SCHA)

This section provides instructions for the Sun Cluster Agent upgrade. It consists of the following sections:

- To Upgrade to the Messaging Server 7 Sun Cluster Agent (MS_SCHA)
- To Upgrade to the Messaging Server 7 Sun Cluster Agent (MS_SCHA) if Cluster Nodes Include Non-Global Zones
- To Upgrade to the Messaging Server 7 Sun Cluster Agent (MS_SCHA) in a Two-node Symmetric Sun Cluster HA Environment

To Upgrade to the Messaging Server 7 Sun Cluster Agent (MS_SCHA)

1. Run `commpkg upgrade` on all nodes on the cluster.
   Messaging Server should be upgraded to 8.0 before upgrading Messaging Server Sun Cluster Agent.
2. Enable Messaging Server resource:

```
# scswitch -e -j <msg_svr_resource>
```

To Upgrade to the Messaging Server 7 Sun Cluster Agent (MS_SCHA) if Cluster Nodes Include Non-Global Zones

If a machine that has non-global zones participates in a cluster, all zones on that machine must be in the cluster. The Sun Cluster software and HA agents should be installed in all zones, and `MS_SCHA` should be installed in the global zone and automatically propagated into all non-global zones (that is, don't use the `-G` switch to `pkgadd`). The Communications Suite Installer treats HA agents like `MS_SCHA` as a product that should be propagated to all non-global zones when it is installed in the global zone. In the rare case where you have managed to install the pre-version 7 `MS_SCHA` agent in the non-global zones, then an upgrade consists of first uninstalling the older agent from all non-global zones, followed by installing the new 7 `MS_SCHA` agent in the global zone.

To check if the older pre-version 7 agent was installed in the global zone and automatically propagated to all non-global zones, verify that `SUNWscims` is listed in `/var/sadm/install/gz-only-packages`. If it is, then run `commpkg upgrade` in the global zone. If it is not listed, then `SUNWscims` is either not installed, or is installed so that it is propagated to non-global zones. If this is this case, use the following procedure:

1. Run `commpkg uninstall` and uninstall `MS_SCHA` in every non-global zone (do not uninstall it in the global zone).
2. In the global zone, run `commpkg upgrade` and upgrade `MS_SCHA`.

To Upgrade to the Messaging Server 7 Sun Cluster Agent (MS_SCHA) in a Two-node Symmetric Sun Cluster HA Environment

1. Upgrade Messaging Server to Version 8.0 before upgrading the Messaging Server Sun Cluster Agent.
2. Make sure that the Messaging Server installation location is accessible from both nodes.
   This is required because a resource type upgrade command validates accessibility. For the first instance in a Symmetric Cluster setup, Messaging Server installation is done on first node only (on a shared storage mount point). For the second instance, Messaging Server installation is done on second node only.
3. Follow the steps mentioned in To Upgrade to the Messaging Server 7 Sun Cluster Agent (MS_SCHA).

> **Note**
> If you prefer to upgrade Sun Cluster Agent (MS_SCHA) for only one instance, then follow the prior steps and correct the resource type version using Sun Cluster commands.

Chapter 11. New Features in Messaging Server 8.0

New Features in Messaging Server 8.0

Messaging Server 8.0 includes the following changes and new features:

- Messaging Server Minor Features
- Platform Support
- Improved Security
- Default Change for the ignoremultipartencoding Channel Option
- BINARYMIME SMTP Extension Supported for Message Submission
- Messaging Server Supports IMAP LIST Extension For Special-Use Mailboxes
- IMAP Append Behavior Change
- Additional Changes to IMAP APPEND
- Support for the MT-PRIORITY SMTP Extension Implemented
- Change in Locks Associated with Transaction Logging
- Additional Functionality of the $, Metacharacter
- Change in Behavior of the Sieve size Test Inside of foreverypart Loops
- Specialized Handling for MX Entries
- New FORWARD Mapping Metacharacters
- New check_memcache.so Mapping Callout
- $T in a LOG_ACTION Mapping Template
- LDAP_DOMAIN_ATTR_CAPTURE MTA Option
- Changes to the Limits Set by the MAX_FILEINTOS, MAX_REDIRECTS, and MAX_ADDHEADERS MTA Options
- Change to Sieve Redirects
- MTA Counters Upgraded from 32 to 64 Bits
- -channel Qualifier Now Accepts Optional List of Channels to Display
- MTA Counters Added to Match Logging for Timers
- New LOG_UID MTA Option
- LOG_MAILBOX_UID MTA Option
- The -iemultipart Qualifier to imsimta test -mime is No Longer the Default.
- Additional Capability Added to imsimta test -rewrite
- Options Added to Specify Attributes to Retrieve During SUBMIT/SMTP Authentication
- Internal lookaside List Increased
- Ability to Access and Manipulate Data Using the memcache Protocol in Sieve
- Ability to Access and Manipulate Data Stored Using MeterMaid in Sieve
- IMAP4 Extension for Returning STATUS Information in LIST Command Response
- IMAP Search ESEARCH RETURN (ALL) processed by the Indexing and Search Server
- Debug for IMAP Search and Sort Command Processing
- IMAP MULTISEARCH Extension
- Removed the -a Switch From the Deliver Command (Incompatible Change)
- Improved configure Behavior With Existing Deployment

Messaging Server Minor Features

See the topic on features introduced in Messaging Server 8.0 in *Messaging Server System Administrator's Guide*.

Platform Support

Messaging Server now supports Solaris 11, Oracle Linux 6.x and Red Hat Enterprise Linux 6.x.

Improved Security

This section includes the improved security features in this release of Messaging Server.

Change to the Default SSL/TLS Cipher Suites

The following cipher suite is no longer enabled by default starting with this release of Messaging Server.

- `SSL_RSA_WITH_RC4_128_MD5`

The following cipher suites are enabled by default starting with this release of Messaging Server:

- `TLS_RSA_WITH_AES_256_CBC_SHA`
- `TLS_RSA_WITH_AES_128_CBC_SHA`

These default changes are the opposite of the defaults in previous releases of Messaging Server. If you are using a mixture of old and new servers, it is recommended you also enable these two cipher suites in Messaging Server 7 Update 5 and prior releases with the `ssladjustciphersuites` option for unified configuration or the `local.ssladjustciphersuites` `configutil` parameter for legacy configuration. Otherwise a slower cipher suite, such as `SSL_RSA_WITH_3DES_EDE_CBC_SHA` may be used when SSL connections are made between versions.

This information is now included in the following logs:

- Protocol log at `info` log level
- Protocol transcript, if enabled
- `msgtrace` log
- POP mailbox status log

The POP log now includes the `authtype` and `auth` session ID.

STARTTLS Option for All LDAP Connections

When the `base.ldaprequiretls` option is set to 1, then connections to LDAP that are not otherwise over LDAPS (port 636) will use the LDAP StartTLS control to negotiate TLS protection. This option is only available in Unified Configuration mode.

Simplification of Enabling SSL

The `enablesslport` option no longer requires the `sslusessl` option to be set explicitly.

New implicitsaslexternal and explicitsaslexternal Channel Options

The `implicitsaslexternal` option on the current source channel causes the SMTP/SUBMIT server to perform an implicit AUTH EXTERNAL SASL operation when a MAIL FROM command is received provided the following conditions have been met:

- `mustsaslserver` is in effect and no authentication operation has been performed.
- An SSL/TLS layer has been successfully negotiated.
- The client provided a valid certificate as part of the SSL/TLS exchange.

  The `explicitsaslexternal` option, the default, disables this behavior.

MeterMaid's Client Now Supports Multiple MeterMaid Servers and SSL for Communication.

MeterMaid's client `check_metermaid.so` now supports multiple MeterMaid servers and SSL for communication.

SSL Support Added to Messaging Server's IMAP Search When Communicating with Indexing and Search Service

SSL support has been added to Messaging Server's IMAP search when it communicates with Indexing and Search Service to send/receive search requests.

TLS Cipher Name Now Uses the Full Standard TLS Cipher Suite Name Instead of the Short Form of the Name

The TLS cipher name included in the application information string will now use the full standard TLS cipher suite name, instead of a short form of the name. This makes logging more informative and provides additional information for includes mappings using this string. However, in the unlikely event customers have written mappings that depend on the abbreviated cipher name, they may need to be updated.

SSLv3 Disabled By Default

The `sslv3enable` option now defaults to 0 instead of 1. This may cause interoperability problems with third party products that have TLS 1.0 disabled by default but have SSL 3.0 enabled. Such products may have security vulnerabilities and may need to be updated for security reasons.

UNAUTHENTICATE Command Disabled by Default

The UNAUTHENTICATE command is now disabled by default. It can be enabled by setting `imap.capability_x_unauthenticate` to 1 (or `service.imap.capability.x_unauthenticate` for legacy configuration).

The `immonitor-access` tool has SSL and SASL support.

Users can add the `-X` switch to enable SASL or the `-T` switch to enable SSL.

NSS version check

The `imsimta version` command now displays the version of NSS installed.

Bundled NSS Upgraded to NSS 3.17.4

This release of Messaging Server upgrades NSS to version 3.17.4. Previously we supported SSL 3.0 and TLS 1.0 only. This adds support for TLS 1.1 and TLS 1.2. There is a new option to enable TLS 1.2 `base.tlsv12enable`. TLS 1.2 is off by default.

Changes to restricted.cnf, the Pipe Channel, and Privileged Shared Libraries.

The following changes are related to Unix user identity that improve product security.

- `restricted.cnf` is now required by default.
- Pipe channel user switching is now disabled by default.

- Privileged shared libraries must be owned by `root` or `bin`.

Security Enhancements to the BURL_ACCESS Mapping Table

$T in a BURL_ACCESS mapping makes use of TLS mandatory for the IMAP connection. $X disables use of TLS. $B in a BURL_ACCESS mapping disables certificate chain of trust validation for IMAPS: URLs and IMAP STARTTLS operations.

Legacy proxyauth Command Now Disabled by Default

There is a new boolean option: `local.legacy_proxyauth` (legacy config) or `imap.legacy_proxyauth` (unified config). This is 0 by default. Set to 1 to re-enable the legacy command. See *Messaging Server System Administrator's Guide* for a detailed discussion of proxy authentication with respect to the MMP. SASL PLAIN is the only supported form of proxy authentication for the MMP and is recommended for other servers. Java Mail clients should use the `mail.imap.sasl.authorizationid` property to perform proxy authentication using SASL PLAIN.

Default Change for the ignoremultipartencoding Channel Option

The `ignoremultipartencoding` channel option is now the default.

BINARYMIME SMTP Extension Supported for Message Submission

The BINARYMIME SMTP extension defined in RFC 3030 is now supported for message submission.

Messaging Server Supports IMAP LIST Extension For Special-Use Mailboxes

Messaging Server now supports the IMAP LIST extension for special-use mailboxes as defined in RFC 6154. This enables compliant mail clients to identify (and label) the folder used for Trash, Drafts and other special uses regardless of the user's language or other name variations.

IMAP Append Behavior Change

The IMAP Append command no longer holds the mailbox lock while receiving a message over the network. This means that problems caused by mailbox locks (such as deferred message delivery) will be less frequent. However, this requires the message to be stored in a staging area and thus append operations will use slightly more I/O than they did previously.

Additional Changes to IMAP APPEND

IMAP APPEND can now reject large messages. A new `maxmessagesize` Unified Configuration option has been added to reject large messages appended to the mailbox. It specifies the maximum message size that IMAP clients are allowed to save via the IMAP APPEND command. The default is 4294967295.

Additional changes to IMAP APPEND will have the following effects:

- Customers will see fewer `mailbox locked` errors that cause delivery delays.
- APPEND will spool messages in transit to a new `append_temp` directory in each partition. If this transfer is interrupted, this will be cleaned up later by `imexpire`. On success it will be hard-linked into the user's mailbox.
- It will be possible to have multiple append commands in progress to the same mailbox. This was not previously possible.

- The I/O cost of doing an IMAP append will increase slightly due to the additional hard-link operation.

Support for the MT-PRIORITY SMTP Extension Implemented

Priority message handling is now configurable through support of the MT-Priority SMTP extension defined in RFC 6710. See the discussion about priority message handling in *Messaging Server System Administrator's Guide* for details.

Change in Locks Associated with Transaction Logging

The locks associated with MTA transaction logging have been moved so they only encompass MTA file operations. In particular, syslog calls are now excluded from these locks.

Increase in the Maximum Size of the Filter Result Logging Field

The maximum size of the filter result logging field in MTA transaction logging (`fl` attribute in XML format) has been increased from 256 to
1024 characters.

Additional Functionality of the $, Metacharacter

A `$,` metacharacter will now expand to the current MTA subaddress character in URL substitutions.

Change in Behavior of the Sieve size Test Inside of foreverypart Loops

The behavior of the Sieve `size` test inside of `foreverypart` loops has been changed. Previously, `size` operated on the message as a whole no matter what the context. Now it operates on the current part only. Note that only decoded part data is considered. Part headers are not included in the size calculation. Also note that the size of non-leaf (message and multipart) parts is currently zero.

This nonstandard extension to the Sieve `size` test is mainly intended to be used to implement attachment size checks. However, since the `size` test can also be used as a function call (in which case it returns the size in octets), this can also be used in conjunction with `foreverypart` to build message manifests for insertion into header fields or logging with the `transactionlog` action.

Specialized Handling for MX Entries

There is specialized handling for MX entries of the form:

```

nomail          IN MX 0     .
```

Such entries are intended to be an indication that host *nomail* does not operate a mail server. Support has been added so that `mailfromdnsverify` will treat such hosts as not being a valid source of mail. Additionally, attempts to send to such a host will fail immediately after the MX lookup instead of attempting any sort of A record lookup.

New FORWARD Mapping Metacharacters

The table below shows the two new `FORWARD` mapping metacharacters and their descriptions.

Metacharacter	Description
$K	Don't reset the intermediate address before processing the mapping/database result. This is useful when performing a final fix up to an address produced by delivery option processing.
$P	Treat the FORWARD mapping result as having specified additional recipient address(es) in addition to rather than replacing the current recipient address.

Both of these metacharacters are no-ops unless $Y and $D are also set.

New check_memcache.so Mapping Callout

A check_memcache.so mapping callout has been developed to allow access to memcache from mappings.

$T in a LOG_ACTION Mapping Template

A $T, if specified in a LOG_ACTION mapping template, causes a tag value to be read from the mapping result. This tag is then prepended to all subsequent LOG_ACTION probes for the same group of log entries. Note that log entry grouping is in general unpredictable, but it is safe to assume that all of the E and D entries associated with a single file in the queue will be in the same group.

LDAP_DOMAIN_ATTR_CAPTURE MTA Option

The LDAP_DOMAIN_ATTR_CAPTURE MTA option can now be used to specify the name of a domain LDAP attribute that will be used to trigger automatic capturing of user or group e-mail messages for all users and groups in the domain. There is no default, no pre-defined LDAP attribute for this purpose.

The value(s) of the LDAP attribute named by LDAP_DOMAIN_ATTR_CAPTURE should be the address(es) to which the captured message copies are supposed to be sent. When a user in the domain has this attribute specified on their LDAP entry, both messages sent to them, as well as from them, will also have a capture copy (normally an encapsulated copy with an entirely new message envelope) sent to the specified address.

The CAPTURE_FORMAT_DEFAUT MTA option controls whether message copies generated due to use of the LDAP attribute named by LDAP_DOMAIN_ATTR_CAPTURE are generated in DSN encapsulated format, or to being in envelope journal format.

Changes to the Limits Set by the MAX_FILEINTOS, MAX_REDIRECTS, and MAX_ADDHEADERS MTA Options

The limits set by the MAX_FILEINTOS, MAX_REDIRECTS, and MAX_ADDHEADERS MTA options now only apply to user-level Sieves.

Change to Sieve Redirects

Sieve redirects now queue to the process rather than the reprocess channel.

MTA Counters Upgraded from 32 to 64 Bits

All MTA counters have been upgraded from 32 to 64 bits. All floating point calculations done on counters are now performed in double precision.

Since there is no practical way to return a 64 bit value in 32 bits, and various counters routinely exceed

32 bits on modern systems, the API routines PMDF_get_channel_counters and PMDFgetChannelCounters have been abandoned. Existing code that calls will not return any results. The new routines are PMDFgetChannelCounters64 and PMDF_get_channel_counters64. Note that apidef.h now contains a

```
#define PMDFgetChannelCounters PMDF_get_channel_counters64
```

So simply recompiling and relinking any code that called the old routine may be sufficient.

-channel Qualifier Now Accepts Optional List of Channels to Display

The -channel qualifier to imsimta counters -show now accepts an optional list of channels to display. Glob-style wildcards can be used in the channel names.

MTA Counters Added to Match Logging for Timers

Additional MTA counters have been added to match the logging for timers described in the "Features Introduced in Messaging Server 8.0" section in *Messaging Server System Administrator's Guide*. The following imsimta counters -show output shows new counters in action.

```
imsimta counters -show -channel=(tcp_local,tcp_internal)

tcp_internal
Received                     0             0             0
Stored                       0             0             0
Delivered                    0             0             0 (0 first
time)
Submitted                   12            14           106
Attempted                    0             0             0
Rejected                     0             0             0
Failed                       0             0             0

Filter[1] failures/used            0/10 = 0.0000
Filter[1] time/used                53.44/10 = 5.3440
Mapping callout time/calls         18.55/5 = 3.7100
FROM_ACCESS callout time/calls     1.07/1 = 1.0700
ORIG_SEND_ACCESS call time/calls   2.48/1 = 2.4800
SEND_ACCESS callout time/calls     4.00/1 = 4.0000
ORIG_MAIL_ACCESS call time/calls   5.00/1 = 5.0000
MAIL_ACCESS callout time/calls     6.00/1 = 6.0000
Transaction time/submitted         52.75/12 = 4.3958
Queue write time/files             5.62/12 = 0.4683

tcp_local
Received                     6             8            27
Stored                       0             0             0
Delivered                    5             7             9 (5 first
time)
Submitted                  225           239          4680
Attempted                    0             0             0
Rejected                   191           193             0
Failed                       1             1             2

Queue time/count                   1825/6 = 304.17
Queue first time/count             1825/6 = 304.17

Filter[1] failures/used            0/225 = 0.0000
Filter[1] time/used                2079.24/225 = 9.2411
Filter[4] failures/used            0/225 = 0.0000
Filter[4] time/used                58.94/225 = 0.2620
Transaction time/submitted         630.19/225 = 2.8008
Queue write time/files             54.15/225 = 0.2407

Current In Associations            2
Total In Associations              1332
Total Out Associations             3
Rejected Out Associations          1
Failed In Associations             127
```

New LOG_UID MTA Option

Certain alias operations, particularly alias expansion of user addresses, involve looking up LDAP entries with UID attributes. When such entries are encountered, the UID is carried through the UID expansion process and, in the case of delivering to the Message Store, the UID is typically incorporated into the resulting address. The LOG_UID MTA option provides the means to log such UIDs. This can be useful when there is a need to identify the last LDAP entry involved in the alias expansion. Note that UIDs are only logged on message enqueue operations. There is no UID available to log on message dequeues.

The LOG_UID MTA option defaults to 0. Setting bit 0 (value 1) logs any available uid. The uid appears immediately after the initial recipient address. A `ui` attribute is used in the XML log format. If bit 1 (value 2) is set in the LOG_UID MTA option, then the UID appears in the LOG_ACTION mapping table probe immediately after the initial destination address field.

LOG_MAILBOX_UID MTA Option

Messages delivered to an IMAP store are tagged with a UID and the folder's UIDVALIDITY value upon insertion. The `LOG_MAILBOX_UID` MTA option provides the means to log this information. At present the field consists of the two values delimited by a colon. This can be useful when there is a need to correlate a message in the store with MTA actions.

The `LOG_MAILBOX_UID` MTA option defaults to 0. Setting bit 0 (value 1) logs the UID and UIDVALIDITY of messages delivered by the ims-ms channel to the store. The UID and UIDVALIDITY appears immediately after the LDAP UID. A `mu` attribute is used in the XML log format. If bit 1 (value 2) is set in the `LOG_MAILBOX_UID` MTA option, then the information appears in the `LOG_ACTION` mapping table probe immediately after the LDAP UID.

The -iemultipart Qualifier to imsimta test -mime is No Longer the Default.

This was changed to match the default product setting to ignore content-transfer-encoding headers on MIME multiparts.

Additional Capability Added to imsimta test -rewrite

For some time `imsimta test -rewrite` has provided a means to enter 8 bit values in text strings by enclosing a series of hexadecimal values in up-arrows, for example,

```
Address: abc^ab ac ad^ghi@domain.com
```

(An up-arrow is specified by doubling, e.g. ^^.)

This works, but is not terribly convenient for entering utf-8 characters. An additional capability has been added to specify Unicode codepoints which are then encoded in utf-8: Simply specify the value as an entity value as in XML:

```
Address: abc&ab;&ad;ghi@domain.com
```

Note that this does NOT produce the same value as the previous example.

(An ampersand can be specified by doubling, e.g., &&.)

Options Added to Specify Attributes to Retrieve During SUBMIT/SMTP Authentication

When authentication occurs in SUBMIT/SMTP various LDAP attributes are retrieved from the authenticated user's LDAP entry. Previously the LDAP attributes used were hard-coded. There are now options that can be used to specify the attribute to retrieve. The following table describes these options, their defaults, and their usage.

Option	Default	Usage
`ldap_attr_auth_sender`	`mail`	Authenticated sender address
`ldap_attr_auth_submit_channel`	`mailSMTPSubmitChannel`	Override source channel for subsequent transactions
`ldap_attr_auth_mail_host`	mailhost	Mail host value used in BURL commands

Internal lookaside List Increased

The size of the internal lookaside list used by the address parser has been increased from 200 elements to 20000 elements. This should limit memory fragmentation resulting from processing large numbers of messages with 10s or 100s of thousands of header addresses.

Ability to Access and Manipulate Data Using the memcache Protocol in Sieve

The ability to access and manipulate data using the `memcache` protocol has been added to Sieve. Access to this mechanism is controlled by the `ENABLE_SIEVE_MEMCACHE` MTA option. This option has three possible values:

0 - `memcache` access disabled
1 - `memcache` access allowed in both user and system level Sieves (default)
2 - `memcache` access only allowed in system-level Sieves

See the discussion about Sieve in *Messaging Server Administration Reference*.

Ability to Access and Manipulate Data Stored Using MeterMaid in Sieve

The ability to access and manipulate data using MeterMaid has been added to Sieve. Access to this mechanism is controlled by the `ENABLE_SIEVE_METERMAID` MTA option. This option has three possible values:

0 - MeterMaid access disabled
1 - MeterMaid access allowed in both user and system level Sieves (default)
2 - MeterMaid access only allowed in system-level Sieves

Three additional options have been added that provide information about how to access the MeterMaid server or servers:

- The `metermaid_host` MTA option specifies the default MeterMaid host for the Sieve metermaid operator. This MTA option if set will override the legacy configuration `metermaid.config.serverhost configutil` parameter, or its Unified Configuration equivalent, the `server_host` MeterMaid MTA client option. There is no default.
- The `metermaid_port` MTA option specifies the MeterMaid port for the Sieve MeterMaid operator. This MTA option if set overrides the legacy configuration `metermaid.config.port configutil` parameter, or its Unified Configuration equivalent, the `port` MeterMaid option. If neither the MeterMaid option nor `configutil` parameter/MeterMaid option is set, then the default is 63837.
- The `metermaid_secret` MTA option specifies the secret string or strings used to verify MeterMaid communications. For the Sieve MeterMaid operator, this MTA option if set overrides

the legacy configuration `metermaid.config.secret configutil` parameter, or its Unified Configuration equivalent, the secret MeterMaid option. There is no default.

See the discussion about Sieve in *Messaging Server Administration Reference*.

IMAP4 Extension for Returning STATUS Information in LIST Command Response

We have implemented the IMAP4 extension for returning STATUS information in the LIST command response as defined in RFC 5819.

IMAP Search ESEARCH RETURN (ALL) processed by the Indexing and Search Server

If the Indexing and Search Server is enabled, the IMAP SEARCH command with the RETURN (ALL) result option will now be sent and processed by the Indexing and Search Server. All other search commands return options will not be sent to the Indexing and Search Server and will be processed by the IMAP server itself. Note that all the other rules of using the Indexing and Search Server still apply. See the `service.imap.indexer.enable` option documentation for more details.

Prior to this change, all SEARCH commands with return options were processed by the IMAP server itself and were not sent to the Indexing and Search Server.

Debug for IMAP Search and Sort Command Processing

We have added a new `search` key to the `debugkeys` option to enable debug about the IMAP search and sort command processing at `DEBUG` level. This will log events noting whether the command is being processed by the Indexing and Search Server or the IMAP server itself, and if the Indexing and Search Server returned an error.

IMAP MULTISEARCH Extension

We now support the `MULTISEARCH` extension defined in RFC 7377. The implementation is fully compliant unless ISS is enabled, in which case the standard ISS restrictions and caveats apply if the ISS subset is used. This introduces two new configuration options: `imap.maxsearchmailboxes` (Unified Configuration) or `local.imap.maxsearchmailboxes` (legacy configuration) and `imap.capability_multisearch` (Unified Configuration) or `service.imap.capability.multisearch` (legacy configuration). See the reference documentation for details about these options. Note that this also adds the `MULTISEARCH` capability to the IMAP capability list.

Removed the -a Switch From the Deliver Command (Incompatible Change)

We have removed the -a switch from the `deliver` command.

Improved configure Behavior With Existing Deployment

The `configure` command has been changed to detect mismatches in certain critical LDAP attributes when performing second and subsequent initial configurations using the same LDAP server. The critical attributes are:

- default domain: `inetDomainBaseDN`, `preferredMailHost`, and `sunPreferredDomain`
- admin user: userPassword, mailHost, and mail

The admin's `userPassword` must match unless the `-novalidate` or `-noldap` options are used with `configure` (in which case the new value will replace the old one when the LDIF generated by `configure` is applied). In interactive mode, the admin may select whether to preserve or replace the other attributes. The default behavior is replace (as with previous versions), but the new `-preserveCritical` option changes the default behavior to preserve. If a state file is used, the default behavior is applied to all attributes except `userPassword`.