Sun™ Internet Mail Server™ 4.0 Release Notes

THE NETWORK IS THE COMPUTER™

A Sun Microsystems, Inc. Business
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 805-7753-10
Revision A, July 1999

Please
Recycle

Adobe PostScript

Contents

CHAPTER 1

Upgrade and Migration Updates

This chapter describes issues or updates for upgrading or migrating from SIMS 3.x to SIMS 4.0.

Note – Go to `http://sun.com/sims/` for updated release notes and other product information concerning the Sun Internet Mail Server 4.0 (SIMS).

Several aspects of SIMS 4.0 differ significantly from the previous SIMS release. In particular, the provisioning model has undergone substantial changes, partly to accommodate the hosted domain features. For information on migrating from SIMS 3.x to SIMS 4.0, please contact your Sun representative or authorized Sun support provider. See `http://sun.com/service/contacting/index.html` for information on contacting Sun and `http://internet.central.sun.com/service/support/index.html` for information on Sun's support services.

Migration Updates

Shared Folder Changes

SIMS no longer keeps a master copy of the shared folder that only the owner can access or update. Instead, each member of the distribution list has her own private copy which can be updated independently, and the owner needs to be a member to access to her own copy. Master copies of shared folders in a SIMS 3.x message store will no longer be accessible once the system is upgraded to SIMS 4.0. These shared folders should be moved or deleted using the `imdeluser` command.

For each shared folder to be removed (the list of these folders can be obtained by listing the contents of `/var/opt/SUNWmail/ims/shared/*`), run the `imdeluser` command using the SIMS Administrator credentials and specifying the shared folder name in place of a user to delete. This will remove the master copy of the shared folder but preserve each member's private copy.

To move a 3.x shared folder:

1. **If the owner was not a member, make the owner a member of the distribution list so she can have her own shared folder copy after the upgrade.**
2. **Locate the UNIX directory containing the master copy of the shared folder:**

   `/var/opt/SUNWmail/ims/shared/*/`*shared_folder_name*
3. **Locate the UNIX directory containing the owner's folders:**

   `/var/opt/SUNWmail/ims/user/*/`*username*
4. **Move (UNIX mv) the master copy under the owner shared space. For example:**

   ```
   % mv /var/opt/SUNWmail/ims/shared/212/sharedfolder/INBOX
   /var/opt/SUNWmail/ims/user/155/owner/shared/sharedfolder
   ```
5. **Remove (UNIX rm) the remaining UNIX directory which contained the master copy For example:**

   ```
   % rm -rf /var/opt/SUNWmail/ims/shared/212/sharedfolder
   ```

`syslog` File Format Changes for Message Store Logging

The `syslog` format for the SIMS 4.0 Message Store and Message Access (MSMA) logging entries has changed since the SIMS 3.5 release. If you have written scripts for parsing the `syslog` file, be aware of these changes so you can update your scripts as well as take advantage of the additional information provided.

For example, the 3.5 MSMA `syslog` format for login messages was as follows:

```
Sep 17 15:11:51 desert imaccessd[<pid>]: Login user=<uid>
host=<hostname>
```

where,

`<pid>` is the process ID of the program issuing the message.

`<uid>` is the user ID of the client.

`<hostname>` is the client hostname string.

The 4.0 MSMA `syslog` format for login messages is now:

```
Apr 29 08:53:49 desert SUNWmail.ims.imaccessd[<pid>]:
<protocol>[<tid>]: Login user=<uid+domain_id> host=<hostid>
```

where,

`imaccessd` changes according to the program issuing the message. For the login/logout messages represented here, the program is `imaccessd`.

`<pid>` is process ID of the program issuing the message.

`<protocol>` may be either pop3, pop3s, imap, imaps depending upon the protocol and use of SSL.

`<tid>` is the thread ID of the thread within the process issuing the message.

`<hostid>` is the client IP address when the `ims_client_lookup` attribute in `ims.cnf` is set to `DNSOFF` (DEFAULT). When `ims_client_lookup` is `DNSON` `<hostid>` is the client hostname string.

CHAPTER 2

Installation Updates

This chapter describes updates and workarounds for the known problems that occur during installation and initial configuration.

Note – Go to `http://sun.com/sims/` for updated release notes and other product information concerning the Sun Internet Mail Server 4.0 (SIMS).

Post Installation Tasks

After you install SIMS 4.0, use the SIMS Administration Console to adjust the `imta dirsync` and `impurge` schedules within `crontab`. `impurge` has no default `crontab` entry and will not run until it is configured in `crontab`.

To configure the purge schedule:

See “Message Purge” in Chapter 7, “Message Store Administration” in the *SIMS 4.0 Administrator’s Guide*.

To reconfigure the alias synchronization schedule:

See “Alias Synchronization Schedule” in Chapter 5, “Internet Mail Transport Agent” in the *SIMS 4.0 Administrator’s Guide*.

Installation Updates

The following section describes any updates limitations to the SIMS 4.0 installation procedures.

Wrong NSDS Location in Installation Guide (bug 4247530)

The *SIMS 4.0 Installation Guide* (Step 2 of "Installing Netscape Directory Services 4.1" of Appendix A) currently refers to the NSDS package as if it was located on the SIMS 4.0 CD. The instructions state "Insert the SIMS CD-ROM into the disk drive."

NSDS is delivered on a separate CD in the SIMS packaging and information about the most current version can be found on the following at `http://sun.com/sims/tech-info/nsds.jhtml`.

setup-tty Allows Installation Over Existing SIMS Setup (bug 4232885)

Description: If you run `setup-tty` after installing SIMS, you are allowed to install over an existing SIMS setup. This should not occur; in fact, you should not be allowed to go past the options page.

Workaround: Automatic upgrade is not supported in this release. Installing over an existing SIMS installation will not save configuration or data.

Install Overwrites Previously saved LDAP Directory (bug 4245234)

Description: Installing SIMS 4.0 can overwrite anything previously saved in the `/var/opt/SUNWconn/ldap` directory. The following steps reproduce the problem:

1. Invoke `uninstall`
2. Select the `do not remove the SDS directory` option
3. Create a tar archive of the `dbm` directory
4. Install SIMS 4.0

The following are the results:

- The `dbm` files in `/var/opt/SUNWconn/ldap/dbm` are deleted
- The tar file in `/var/opt/SUNWconn/ldap` is deleted

Workaround: There is currently no workaround. Suggest saving the tar file into a directory other than `/var/opt/SUNWconn/ldap`.

Schema Files for NSDS slapd.conf Removed with Uninstall

Description: Performing an uninstall of SIMS 4.0 removes the SIMS schema files included in the NSDS `slapd.conf` file. The following steps reproduce this problem:

1. Install NSDS 4.1.
2. Edit `/usr/netscape/server4/slapd-<hostname>/config/slapd.conf` to add the following four lines:

```
include "/usr/netscape/server4/slapd-minister/config/sims-sisp.at.conf"
include "/usr/netscape/server4/slapd-minister/config/sims-sisp.oc.conf"
include "/usr/netscape/server4/slapd-minister/config/sims.at.conf"
include "/usr/netscape/server4/slapd-minister/config/sims.oc.conf"
```

3. Install SIMS 4.0.
4. Uninstall SIMS 4.0 with the `uninstall -d sims` command

The four lines added to the `slapd.conf` file disappears.

Workaround: Manually add the four lines back into the `slapd.conf` file.

CHAPTER 3

Software Limitations

This chapter describes known software limitations and updates for the SIMS 4.0 product.

Note – Go to `http://sun.com/sims/` for updated release notes and other product information concerning the Sun Internet Mail Server (SIMS) 4.0.

Upgrading Solaris

The following steps should be performed after upgrading from Solaris 2.6 to Solaris 7 (and after installing SIMS 4.0):

1. **Add the** `libldap` **patch for Solaris 7 (107555 for SPARC / 107556 for Intel).**
2. **Rename** `/usr/lib/sendmail` **to** `/usr/lib/sendmail.orig`**:**

```
# mv /usr/lib/sendmail /usr/lib/sendmail.orig
```

3. **Create a symbolic link from** `/opt/SUNWmail/imta/bin/sendmail` **to** `/usr/lib/sendmail`**:**

```
# ln -s /opt/SUNWmail/imta/bin/sendmail /usr/lib/sendmail
```

Runtime Limitations

This section addresses the issues when SIMS 4.0 is running. Specifically, these bugs may be found when using the Administration Console, with particular command line utilities, files, or at runtime in general.

Mail Host name needed when Adding Users of Hosted Domain

When adding users under a hosted domain, the name of the mail host must be provided.

For example, a user (`jdoe`) needs to be added to the hosted domain `stream.com`. The mail host for `stream.com` is `bridge.net` on host name `alpha`. When adding `jdoe` with the `imadmin add user` command, the following is a portion of what is entered:

```
# imadmin add user ... -H alpha.bridge.net -n stream.com
```

The `-H` option in the above example should be the mail host name (`alpha.bridge.net`) and not the hosted domain name (`alpha.stream.com`).

The same should be specified when adding a user via the SIMS Admin Console.

imadmin Commands Ignore -d Option (bug 4228418)

Description: The `imadmin` [`add` | `delete` | `modify` | `purge`] `user` commands incorrectly ignore the `-d` option and successfully add users to the domain referred to by the `-n` option.

Workaround: There is no workaround. The command succeeds when it should actually fail.

imadmin Commands Ignore -d Option (bug 4228427)

Description: The `imadmin [add | delete | modify | purge] group` commands incorrectly ignore the `-d` option and successfully add groups to the domain referred to by the `-n` option.

Workaround: There is no workaround. The command succeeds when it should actually fail.

Configuration Variables Not Supported by GUI (bug 4224162)

Description: The following configuration variables in the `ims.cnf` configuration file are not supported in the Admin Console:

- `ims-auth-timeout`
- `ims-md5auth-enable`
- `ims-popb4smpt-timeout`
- `ims-popb4smtp-lib`
- `ims-client-lookup`

Workaround: Use the `imedit` utility to edit the `ims.cnf` configuration file in order to change the parameter values.

User Manager Browser Fails to Show Node on OSI Tree (bug 4243055)

Description: The DIT browser in the Admin Console does not display the `People,Groups` node for a domain that is in the OSI style. The `People` and `Groups` cannot be displayed because the `Find` and `Display All` buttons require either `People` or `Groups` to be selected.

Workaround: Use the `Choose Domain to Browse` field and enter the newly created domain. When the domain is displayed in the mail directory window, `People` and `Groups` are displayed.

Directory Data Locking Does Not Occur (bug 4244578)

Description: When a user entry is concurrently modified (by using both the Admin Console and the CLI tools) the modification is allowed when it should not be allowed.

Workaround: Avoid running the Admin Console and the CLI concurrently on the same data.

Install Changes Needed for autoreply (bug 4244880)

Description: The new autoreply channel is visible and configurable through the Admin Console. However, this channel is strictly for future use. Changes made to this channel (via the Admin Console or manual editing) will not have an effect on the autoreply program in this release of SIMS.

Workaround: No workaround exists.

Directory Services Limitations

This section describes limitations relating to the directory and directory services.

Long Distribution List Name Causes dsservd to Core Dump (bug 4213256)

Description: Adding a long distribution list name (approximately 75 characters or more) can crash the directory server.

Workaround: Do not add distribution lists with names longer than 75 characters.

Text In the Directory

All text data stored in the directory, such as user names, group descriptions, company names, etc., must be in ASCII. It is not guaranteed that anything besides 7-big ASCII will work.

IMTA Limitations

This section describes limitations relating to the IMTA.

Distribution Lists Do Not Support Multi-valued Attributes (bug 4243461)

Description: Distribution lists do not support some multi-valued attributes such as `moderator` and `requests-to`. It retrieves and recognizes only the first value it fins for these attributes.

Workaround: No workaround exists.

Purging Log Files (bug 4247316)

SIMS IMTA log files must be periodically examined and purged or they will continue to grow and fill up your disk. The IMTA log files are located in `/var/opt/SUNWmail/imta/log`.

Native Delivery Not Set to /var/mail Does Not Work (bug 4247441)

Description: If `maildeliveryoption` is native but the `mailmessagestore` attribute is not set to `/var/mail`, mail delivery does not work. Also, if `mailmessagestore` is set to `/var/mail` instead of `/var/mail/` delivery does not work.

Workaround: If the `mailmessagestore` attribute is set to `/var/mail`, change it to `/var/mail/`. If the mailmessagestore attribute is set to something else, add a new rewrite rule for the pipe channel section in the `imta.cnf` file:

```
pipe.mailhost.domain $E$U@pipe-daemon
```

Message Store and Message Access Limitations

This section describes limitations relating to the message store.

I18N Search Bug with ISO-2022-JP (bug 4232992)

An `iconv` bug exists which causes `imaccessd` to crash when searching for ISO-2022-JP email messages.

Message Store Utilities and Virtual Domains

All message store utilities will operate on the entire message store by default. Some of the message store utilities have the added option to specify only a specific domain on which to operate. If the SIMS administrator forgets to specify the desired domain on these utilities, they will target or operate on the entire message store.

imcheck Utility

The options for the `imcheck` utility have changed between SIMS 3.5 and SIMS 4.0. Please see the man page (`imcheck`(1M)) for more information. One difference is with the user test option (`-u`). The user test previously only ran when the message store was down, and would attempt to fix any corrupted folders it found. Now the user test by default runs in a read-only mode This does not fix any folders, but can be run while the message store is up. If the old version of the user test is desired, the SIMS administrator should run imcheck with the `-u` and `-w` options. This requires the message store to be down.

Delegated Management Console Limitations

This section describes the open issues for the Delegated Management Console.

Users Cannot Select Alternate Delivery Programs (bug 4209019)

In previous SIMS releases, the `emailuser.html` page allowed the user to set alternate delivery program options. This feature does not exist in SIMS 4.0.

Online Help for Login ID Not Found on Create Distribution List Page (bug 4245967)

Description: An error (Error 404: Not Found) results when you select online help for the `Login ID` field within the Search area on the Create Distribution List page.

Workaround: No workaround exists. The Edit User page correctly displays the online help for the `Login ID` field, if the exact help text is desired.

Securing Passwords

Password information is passed from the user to the directory and back through a series of programs. Both the user and ISP need to know the user's password. The password's transmission path is:

1. Web browser to HTTP server
2. HTTP server to CGI program
3. CGI program to Delegated Management server
4. Delegated Management server to the directory

A default setup will send this password in plain text along the entire transmission path from the web browser to the directory. This means that the password can be viewed “as is” during transmission. In order to make the password secure, each transmission path needs to be addressed. The following methods can be used to secure the password (and all the other data):

- Use the SSL facility for the transmission between the web browser and the HTTP server.
- The path between the HTTP server and the CGI program is not at risk because information is posted.
- The transmission between the CGI program, the Delegated Management server, and the directory may be protected by an IP-based protection method.

Web Access Limitations

This section describes the open issues / workarounds for the Web Access client.

Dependencies

WebAccess 1.2 depends on the following publicly available standard extensions to Java:

- Java Naming and Directory Interface (JNDI) API version 1.1.1
- JavaMail API version 1.1.1

These standard extensions are included with the WebAccess 1.2 release.

im.server Does Not Start WebAccess

`/etc/init.d/im.server` starts the `sws_server` for the SIMS Admin Console, but not for WebAccess. The WebAccess instance off the Sun Web Server needs to be started manually if WebAccess is installed but the system is not rebooted.

To start WebAccess, execute the following:

```
#/usr/bin/htserver start WebAccess
```

Login Attempts Fail for Valid User (bug 4244818, 4244824, 4244831)

Description: Logging into WebAccess with a valid user account using the HotJava browser can appear fail or produce a blank page.

Workaround: Click the `Reload` button on the web browser or attempt to login again.

Able to Access Account After Exit (bug 4244832)

Description: Using the HotJava browser, it is possible to access a user account after exiting by clicking the `Back` button.

Workaround: No workaround exists.

Internationalization / Localization Limitations

This section describes internationalization / localization issues and limitations.

Errors in enable.I18N.txt (bug 4246892)

In the `enable.I18N.txt` file, the lines under the entry 3 contains errors:

```
# cd <2-letter-localename1>
# ln -s ../../en/images
# ln -s ../../en/help

# cd ../../<2-letter-localename2>
# ln -s ../../en/images
# ln -s ../../en/help
```

The lines should be changed to:

```
# cd <2-letter-localename1>
# ln -s ../en/images
# ln -s ../en/help

# cd ../<2-letter-localename2>
# ln -s ../en/images
# ln -s ../en/help
```

The lines in Step 3 of the example contains errors:

```
Step 3
# cd ja
# ln -s ../../en/images
# ln -s ../../en/help
# cd ../../fr
# ln -s ../../en/images
# ln -s ../../en/help
# cd ../../ru
# ln -s ../../en/images
# ln -s ../../en/help
```

The lines should be changed to:

```
Step 3
# cd ja
# ln -s ../en/images
# ln -s ../en/help
# cd ../fr
# ln -s ../en/images
# ln -s ../en/help
# cd ../ru
# ln -s ../en/images
# ln -s ../en/help
```

CHAPTER 4

Documentation Changes

This chapter describes any updates or changes to the SIMS 4.0 FCS documentation.

Note – Go to `http://sun.com/sims/` for updated release notes and other product information concerning the Sun Internet Mail Server (SIMS) 4.0.

SIMS 4.0 Installation Guide

This section describes any changes or updates to the *SIMS 4.0 Installation Guide*.

Wrong NSDS Location (bug 4247530)

The *SIMS 4.0 Installation Guide* (Step 2 of "Installing Netscape Directory Services 4.1" of Appendix A) currently refers to the NSDS package as if it was located on the SIMS 4.0 CD. The instructions state "Insert the SIMS CD-ROM into the disk drive."

NSDS is delivered on a separate CD in the SIMS packaging and information about the most current version can be found on the following at `http://sun.com/sims/tech-info/nsds.jhtml`.

Errors In Installation Guide

Three errors exist in Appendix B "Installing Netscape Directory Services for SIMS High Availability":

1. Step 2 of “Setting up the Netscape Directory Services Administration Server for SIMS/HA” of Appendix B contains an error. It reads:

   Change the `nsserveraddress` to the logical address for the system.

```
# <shared-file-system>/NSDS/shared/bin/ldapmodify -p <portnumber>
-D "cn=<Directory Manager>" -w <PASSWD>

dn: :cn=configuration, cn=admin-serv-<ha-logical-hostname>,
cn=Netscape Administration Server, cn=Server Group, cn=<ha-
logical-hostname> ou=<root domain name> o=NetscapeRoot
```

The logical address should be:

```
dn: cn=configuration, cn=admin-serv-<ha-logical-hostname>,
cn=Netscape Administration Server, cn=Server Group,
cn=<ha-logical-hostname>, ou=<root domain name>, o=NetscapeRoot
```

2. Step 6 of “Guidelines for Installing and Configuring Sun Cluster and High Availability” of Appendix B contains an error. It reads: “You now need to edit the `nsldap_svc_stop` script on both nodes.”

   The instruction should read: “You now need to edit the `/opt/SUNWcluster/ha/nsldap/nsldap_svc_stop` script on both nodes.”

3. Step 5 of “Registering the Netscape Directory Service with the High Availability Framework” of Appendix B contains an error. It reads:

```
# /opt/SUNWhadf/bin/hareg -r Sun_Internet_Mail -b /opt/SUNWimha/
clust_proga -m START_NET=imha_start_net, STOP_NET=imha_stop_net -
t START_NET=120,STOP_NET=30 pv 4.0 -d NSDS
```

   “pv” should be -v. The command should be:

```
# /opt/SUNWhadf/bin/hareg -r Sun_Internet_Mail -b /opt/SUNWimha/
clust_proga -m START_NET=imha_start_net, STOP_NET=imha_stop_net -
t START_NET=120,STOP_NET=30 -v 4.0 -d NSDS
```

SIMS 4.0 Administrator's Guide

This section describes any changes or updates to the *SIMS 4.0 Administrator's Guide.*

Chapter 11, SIMS Periodic Maintenance Procedures

The following section should be added to Page 232 under IMTA Maintenance:

Purging IMTA Log Files

SIMS IMTA log files must be periodically examined and purged or they will continue to grow and fill up your disk. The IMTA log files are located in `/var/opt/SUNWmail/imta/log`.

SIMS 4.0 Reference Manual

This section describes any changes or updates to the *SIMS 4.0 Reference Manual*.

backup-groups.cnf Configuration File

The `backup-groups.cnf` configuration file is not documented in the *SIMS 4.0 Reference Manual*. The `backup-groups.cnf` file contains definitions for the SIMS Message Store backup groups. The group definitions are used by the message store utilities `imbackup` and `mkbackupdir` to back up users by group. See the UNIX man page (`backup-groups.cnf`(4)) for more information on this configuration file.

References to spmProgramNumber

In Chapter 4, "SIMS Configuration Files," references to the `spmProgramNumber` parameter in the `sims.cnf`(4) file should be ignored. It is not used by the Delegated Management component.

IMTA Option File Options

A list of the IMTA options appears in Chapter 2, “IMTA Configuration,” of the *SIMS 4.0 Reference Manual*. The following sections list each option with a more detailed description.

General Configuration Format Options.

These options affect and modify various aspects of IMTA configuration file format and configuration settings.

COMMENT_CHARS (integer list {33, 59})

`COMMENT_CHARS` controls which characters are taken to signal a comment when they appear in the first column of IMTA input files. The value of this option takes the form of a list of ASCII character values in decimal. The default is the list {33, 59}, which specifies exclamation points and semicolons as comment introduction characters.

EXPROUTE_FORWARD (integer 0 or 1)

`EXPROUTE_FORWARD` controls the application of the `exproute` channel keyword to forward-pointing (To:, Cc:, and Bcc: lines) addresses in the message header. A value of 1 is the default and specifies that `exproute` should affect forward-pointing header addresses. A value of 0 disables the action of `exproute` on forward-pointing addresses.

ID_DOMAIN (string)

`ID_DOMAIN` specifies the domain name to use when constructing message IDs. By default, the official host name of the local channel is used.

IMPROUTE_FORWARD (integer 0 or 1)

`IMPROUTE_FORWARD` controls the application of the `improute` channel keyword to forward-pointing (To:, Cc:, and Bcc: lines) addresses in the message header. A value of 1 is the default and specifies that `improute` should affect forward-pointing header addresses. A value of 0 disables the action of `improute` on forward-pointing addresses.

MAX_ALIAS_LEVELS (integer)

MAX_ALIAS_LEVELS controls the degree of indirection allowed in aliases, that is, how deeply aliases may be nested, with one alias referring to another alias, and so on. The default value is 10.

MISSING_RECIPIENT_POLICY (integer)

RFC 822 (Internet) messages are required to contain a recipient header: a To:, Cc:, or Bcc: header. A message without a header is illegal. However, some broken user agents and mailers (For example, many older versions of sendmail) will emit such illegal messages. MISSING_RECIPIENT_POLICY takes an integer value specifying the approach to use for such messages. The default value, if the option is not explicitly present, is 0, meaning that envelope To: addresses are placed in a To: header.

Value	Action
0	Place envelope To: recipients in a To: header.
1	Pass the illegal message through unchanged.
2	Place envelope To: recipients in a To: header.
3	Place all envelope To: recipients in a single Bcc: header.
4	Generate a group construct To: header, To: Recipients not specified;.
5	Generate a blank Bcc: header.
6	Reject the message.

RECEIVED_DOMAIN (string)

RECEIVED_DOMAIN sets the domain name to use when constructing Received: headers. By default, the official host name of the local channel is used.

REVERSE_ENVELOPE (0 or 1)

REVERSE_ENVELOPE controls whether IMTA applies address reversal to envelope From: addresses as well as header addresses. This option will have no effect if USE_REVERSE_DATABASE is set to 0 or if neither the reverse database nor a REVERSE mapping exist. The default is 1, which means IMTA will attempt to apply any address reversal to envelope From: addresses. A value of 0 will disable this use of the address reversal database and REVERSE mapping.

USE_ALIAS_DATABASE (0 or 1)

USE_ALIAS_DATABASE controls whether IMTA uses the alias database as a source of system aliases for local addresses. The default is 1, which means that IMTA will check the database if it exists. A value of 0 will disable this use of the alias database.

USE_DOMAIN_DATABASE (0 or 1)

USE_DOMAIN_DATABASE controls whether IMTA uses the domain database as a source of rewrite rules. The default is 1, which means that IMTA will check the database if it exists. A value of 0 will disable this use of the domain database.

USE_FORWARD_DATABASE (integer)

USE_FORWARD_DATABASE whether IMTA uses the forward database. This value is a decimal integer representing a bit-encoded integer, the interpretation of which is given in the table below.

Bit	Value	Usage
0	1	When set, the forward database is used.
3	8	When set, channel-level granularity is used with the forward database entries. Forward database entries' left sides must have the form: `source-channel|from-address|to-address` Note the vertical bars, |.
4	16	When set, channel-level granularity is used with the FORWARD mapping. FORWARD mapping entries' patterns left sides must have the form: `source-channel|from-address|to-address` Note the vertical bars, |.

Bit 0 is the least significant bit.

The default value for USE_FORWARD_DATABASE is 0, which means that IMTA will not use the forward database. Note that a FORWARD mapping, if present, is always consulted.

USE_PERSONAL_ALIASES (0 or 1)

USE_PERSONAL_ALIASES controls whether IMTA uses personal alias databases as a source of aliases for local addresses. The default is 1, which means that IMTA will check such databases, if they exist. A value of 0 will disable personal aliases and make them unavailable to all users.

USE_REVERSE_DATABASE (0--255)

USE_REVERSE_DATABASE controls whether IMTA uses the address reversal database and REVERSE mapping as a source of substitution addresses. This value is a decimal integer representing a bit-encoded integer, the interpretation of which is given in the table below.

Bit	Value	Usage
0	1	When set, address reversal is applied to addresses after they have been rewritten by the IMTA address rewriting process.
1	2	When set, address reversal is applied before addresses have had IMTA address rewriting applied to them.
2	4	When set, address reversal will be applied to all addresses, not just to backwards-pointing addresses.
3	8	When set, channel-level granularity is used with the REVERSE mapping. REVERSE mapping table (pattern) entries must have the form: `source-channel|destination-channel|address` Note the vertical bars, |.
4	16	When set, channel-level granularity is used with address reversal database entries. Reversal database entries' left sides must have the form: `source-channel|destination-channel|address` Note the vertical bars, |.
5	32	Apply REVERSE mapping even if a reverse database entry has already matched.
6	64	Apply address reversal to message ids.
7	128	When set, this modifies the effect of bit 4 (channel-level granularity of address reversal database entries); when this bit is also set, the address reversal database entries take the form: `destination-channel|address` Note the vertical bars, |.

Bit 0 is the least significant bit.

The default value for USE_REVERSE_DATABASE is 5, which means that IMTA will reverse Envelope From: addresses and both backwards and forwards pointing addresses after they have passed through the normal address rewriting process. Simple address strings are presented to both the REVERSE mapping and the reverse database. Note that a value of 0 disables the use of the address reversal completely.

Note that the default of 5 represents a change from earlier versions of IMTA in which this option had a default value of 1 (reverse only backwards pointing addresses).

Notification Messages and Jobs Options

These options affect notification messages and the IMTA periodic return job.

ACCESS_ERRORS (integer 0 or 1)

IMTA provides facilities to restrict access to channels on the basis of the NETMBX privilege or on the basis of group ids on UNIX. If ACCESS_ERRORS is set to 0 (the default), when an address causes an access failure IMTA will report it as an “illegal host or domain” error. This is the same error that would occur if the address was illegal. This usage provides an important element of security in circumstances where information about restricted channels should not be revealed. Setting ACCESS_ERRORS to 1 will override this default and provide a more descriptive error.

HISTORY_TO_RETURN (1-200)

HISTORY_TO_RETURN controls the number of delivery attempt history records are included in returned messages. The delivery history provides some indication of how many delivery attempts were made and in some cases indicates the reason the delivery attempts failed. The default value for this option is 20.

LINES_TO_RETURN (integer)

LINES_TO_RETURN controls the number of lines of message content IMTA includes when generating a notification message for which it is appropriate to return only a sample of the contents. The default is 20. Note that this option is irrelevant when generating a NOTARY bounce message, where either the full content or only headers are included, according to the choice specified during the initial submission of the message. In practice, this option is mainly relevant to the warning messages the IMTA return job sends about messages awaiting further delivery retries in the IMTA queue area.

RETURN_ADDRESS (string)

RETURN_ADDRESS sets the return address for the local Postmaster. The local Postmaster's address is *postmaster@localhost* by default, but it can be overridden with the address of your choice. Care should be taken in the selection of this address --- an illegal selection may cause rapid message looping and pile-ups of huge numbers of spurious error messages.

RETURN_DELIVERY_HISTORY (0 or 1)

The RETURN_DELIVERY_HISTORY flag controls whether a history of delivery attempts is included in returned messages. The delivery history provides some indication of how many delivery attempts were made and in some cases indicates the reason the delivery attempts failed. A value of 1 enables the inclusion of this information and is the default. A value of 0 disables return of delivery history information. HISTORY_TO_RETURN controls how much history information is actually returned.

RETURN_ENVELOPE (integer)

RETURN_ENVELOPE takes a single integer value, which is interpreted as a set of bit flags. Bit 0 (value = 1) controls whether return notifications generated by IMTA are written with a blank envelope address or with the address of the local postmaster. Setting the bit forces the use of the local postmaster address, clearing the bit forces the use of a blank address. (The use of a blank address is mandated by RFC 1123.) However, some systems do not handle blank envelope From: addresses properly and may require this option. Bit 1 (value = 2) controls whether IMTA replaces all blank envelope addresses with the address of the local postmaster. This is used to accommodate incompliant systems that don't conform to RFC 821, RFC 822, or RFC 1123. Note also that you can use the returnenvelope channel keyword to impose this sort of control on a per-channel basis.

RETURN_PERSONAL (string)

RETURN_PERSONAL specifies the personal name to use when IMTA generates postmaster messages, for example, bounce messages. By default, IMTA uses the string IMTA e-Mail Interconnect.

RETURN_UNITS (0 or 1)

The time units used by the message return system is controlled using `RETURN_UNITS`; that is, this option controls the interpretation of the values specified for the `notices` keyword. A value of 0 selects units of days; a value of 1 selects units of hours. By default, units of days are used. On UNIX systems, the scheduling of the execution of the message return job is performed by changing the crontab entry controlling when it runs.

USE_ERRORS_TO (0 or 1)

`USE_ERRORS_TO` controls whether IMTA uses the information contained in Errors-to: header lines when returning messages. Setting this option to 1 directs IMTA to make use of this header line. A value of 0, the default, disables this header line. This default represents a change from the default in previous versions of IMTA.

USE_WARNINGS_TO (0 or 1)

`USE_WARNINGS_TO` controls whether IMTA uses the information contained in Warnings-to: header lines when returning messages. Setting this option to 1 directs IMTA to use these header lines. The default is 0, which disables this header line. This default represents a change from the default in previous versions of IMTA.

Message Size Options

These options relate to message size, such as limits on the size of messages allowed in IMTA, message size affecting message processing priority, limits on the extent to which IMTA looks into messages of complex MIME structure, and fine tuning of message fragmentation.

BLOCK_LIMIT (integer > 0)

`BLOCK_LIMIT` places an absolute limit on the size, in blocks, of any message which may be sent or received with IMTA. Any message exceeding this size will be rejected. By default, IMTA imposes no size limits. Note that you can use the `blocklimit` channel keyword to impose limits on a per-channel basis. The size in bytes of a block is specified with `BLOCK_SIZE`.

BLOCK_SIZE (integer > 0)

IMTA uses the concept of a “block” in several ways. For example, the IMTA log files (resulting from placing the `logging` keyword on channels) record message sizes in terms of blocks. Message size limits specified using the `maxblocks` keyword are also in terms of blocks. Normally a IMTA block is equivalent to 1024 characters. Use this option to modify this sense of what a block is.

Note – IMTA stores message sizes internally as an integer number of blocks. If the size of a block in bytes is set to a very small value, it is possible for a very large message to cause an integer overflow. A message size of greater than 2**31 blocks would be needed, but this value is not inconceivable if the block size is small enough.

BOUNCE_BLOCK_LIMIT (integer)

`BOUNCE_BLOCK_LIMIT` can be used to force bounces of messages over the specified size to return only the message headers, rather than the full message content.

LINE_LIMIT (integer)

`LINE_LIMIT` places an absolute limit on the overall number of lines in any message that might be sent or received with IMTA. Any message exceeding this limit will be rejected. By default, IMTA imposes no line count limits. You can use the `linelimit` channel keyword to impose limits on a per-channel basis.

MAX_HEADER_BLOCK_USE (real number between 0 and 1)

`MAX_HEADER_BLOCK_USE` controls which fraction of the available message blocks can be used by message headers.

MAX_HEADER_LINE_USE (real number between 0 and 1)

`MAX_HEADER_LINE_USE` controls what fraction of the available message lines can be used by message headers.

MAX_INTERNAL_BLOCKS (integer)

MAX_INTERNAL_BLOCKS specifies how large (in IMTA blocks) a message IMTA will keep entirely in memory; messages larger than this size will be written to temporary files. The default is 10. For systems with lots of memory, increasing this value may provide a performance improvement.

MAX_MIME_LEVELS (integer)

MAX_MIME_LEVELS specifies the maximum depth to which IMTA should process MIME messages. The default is 100, meaning that IMTA will process up to one hundred levels of message nesting. Higher values may require additional amounts of memory and, for the Dispatcher, additional per-thread storage space.

MAX_MIME_PARTS (integer)

MAX_MIME_PARTS specifies the maximum number of MIME parts that IMTA should process in a MIME message. The default value is 0, meaning no limit is imposed.

NORMAL_BLOCK_LIMIT (integer)

NORMAL_BLOCK_LIMIT can be used to instruct IMTA to downgrade the priority of messages based on size: messages above the specified size will be downgraded to non-urgent priority. This priority, in turn, may affect whether the message is processed immediately, or whether it is left to wait for processing until the next periodic job runs. The value is interpreted in terms of IMTA blocks, as specified by BLOCK_SIZE. You can use the normalblocklimit channel keyword to impose such downgrade thresholds on a per-channel basis.

NON_URGENT_BLOCK_LIMIT (integer)

NON_URGENT_BLOCK_LIMIT may be used to instruct IMTA to downgrade the priority of messages based on size: messages above the specified size will be downgraded to lower than non-urgent priority, meaning that they will not be processed immediately and will wait for processing until the next periodic job runs. The value is interpreted in terms of IMTA blocks, as specified by BLOCK_SIZE. You can use the nonurgentblocklimit channel keyword to impose such downgrade thresholds on a per-channel basis.

URGENT_BLOCK_LIMIT (integer)

URGENT_BLOCK_LIMIT may be used to instruct IMTA to downgrade the priority of messages based on size: messages above the specified size will be downgraded to normal priority. This priority, in turn, may affect whether the message is processed immediately, or whether it is left to wait for processing until the next periodic job runs. The value is interpreted in terms of IMTA blocks, as specified by BLOCK_SIZE. You can use the urgentblocklimit channel keyword to impose such downgrade thresholds on a per-channel basis.

Logging and Counters Options

The options listed in this section affect IMTA logging. LOG_DELAY_BINS and LOG_SIZE_BINS relate to IMTA counters binning. The rest of these logging options affect the formatting of the IMTA log file and logging of optional additional information.

LOG_CONNECTION (integer)

LOG_CONNECTION controls whether or not connection information, for example, the domain name of the SMTP client sending the message, is saved in the mail.log file. This value is a decimal integer representing a bit-encoded integer, the interpretation of which is given in the table below.

Bit	Value	Usage
0	1	When set, connection information is included in E, D and R log records.
1	2	When set, connection open/close/fail records are logged by message enqueue and dequeue agents such as the SMTP and X.400 clients and servers.
2	4	When set, I records are logged recording ETRN events.

For example, enabling LOG_CONNECTION=3 will result both in additional sorts of log file entries—entries showing when an SMTP connection is opened or closed—and additional information in regular log file entries showing the name of the system connecting (or being connected to), or the channel host name of the enqueuing channel when the enqueuing channel is not an SMTP channel. TCP/IP channels have a channel level option that can override this setting for particular channels.

LOG_DELAY_BINS (comma-separated list of up to five integers)

This option specifies the bin divisions for the IMTA counters tracking numbers of messages delivered in the specified number of seconds. The defaults values are 60, 600, 6000, 60000, 600000.

LOG_FILENAME (0 or 1)

`LOG_FILENAME` controls whether the names of the files in which messages are stored are saved in the `mail.log` file. A value of 1 enables file name logging. When file name logging is enabled, the file name will appear as the first field after the final form envelope To: address. A value of 0 (the default) disables file name logging.

LOG_FORMAT (1, 2, or 3)

`LOG_FORMAT` controls formatting options for the `mail.log` file. A value of 1 (the default) is the standard format. A value of 2 requests non-null formatting: empty address fields are converted to the string "< >". A value of 3 requests counted formatting: all variable length fields are preceded by "N:", where "N" is a count of the number of characters in the field.

LOG_HEADER (0 or 1)

`LOG_HEADER` controls whether IMTA writes message headers to the `mail.log` file. A value of 1 enables message header logging. The specific headers written to the log file are controlled by a site-supplied `log_header.opt` file. The format of this file is that of other IMTA header option files. For instance, a `log_header.opt` file containing

```
To: MAXIMUM=1
From: MAXIMUM=1
Defaults: MAXIMUM=-1
```

would result in writing the first To: and the first From: header per message to the log file. A value of 0 (the default) disables message header logging.

LOG_LOCAL (0 or 1)

LOG_LOCAL controls whether the domain name for the local host is appended to logged addresses that don't already contain a domain name. A value of 1 enables this feature, which is useful when logs from multiple systems running IMTA are concatenated and processed. A value of 0, the default, disables this feature.

LOG_MESSAGE_ID (0 or 1)

LOG_MESSAGE_ID controls whether message IDs are saved in the mail.log file. A value of 1 enables message ID logging. When message ID logging is enabled, the message ID will be logged after the final form envelope To: address entry---and after the message file name, if LOG_FILENAME=1 is also enabled. A value of 0 (the default) disables message ID logging.

LOG_NOTARY (0 or 1)

LOG_NOTARY controls whether IMTA includes an indicator of NOTARY (delivery receipt) flags in the mail.log file entries. A value of 1 enables NOTARY flag logging. A value of 0 (the default) disables it. The NOTARY flags will be logged as a bit encoded integer after the current form of the envelope To: address.

LOG_PROCESS (0 or 1)

LOG_PROCESS controls whether the id of the process that enqueues mail is saved in the mail.log file. A value of 1 enables process id logging. A value of 0 (the default) disables it. The process id will be logged in a hexadecimal representation, after the date and time stamps in log entries—and after the node name, if LOG_NODE=1 is also enabled.

LOG_SIZE_BINS (comma-separated list of up to five integers)

LOG_SIZE_BINS specifies the bin divisions for the IMTA counters tracking numbers of messages of the specified number of (IMTA) blocks. The default values are 2, 10, 50, 100, 500.

LOG_SNDOPR (0 or 1)

LOG_SNDOPR controls the production of syslog messages (UNIX) by the IMTA message logging facility. If this feature is enabled by specifying a value of 1, the logging facility will produce a message if it encounters any difficulty writing to the log file. A value of 0 (the default) turns off these messages.

LOG_USERNAME (0 or 1)

LOG_USERNAME controls whether the user name associated with a process that enqueues mail is saved in the mail.log file. A value of 1 enables username logging. When user name logging is enabled, the user name will be logged after the final form envelope To: address field in log entries—and after the message ID, if LOG_MESSAGE_ID=1 is also enabled. A value of 0 (the default) disables user name logging.

SEPARATE_CONNECTION_LOG (0 or 1)

SEPARATE_CONNECTION_LOG controls whether the connection log information generated by setting LOG_CONNECTION=1 is stored in the usual IMTA message logging files, mail.log*, or stored separately in connection.log* files. SEPARATE_CONNECTION_LOG=0, the default, causes connection logging to be stored in the regular message log files; a value of 1 causes the connection logging to be stored separately.

Message Loop Detection and HELD Messages

These options relate to IMTA's facility to sideline as .HELD messages that appear to be looping.

HELD_SNDOPR (0 or 1)

HELD_SNDOPR controls the production of syslog messages (UNIX) when a message is forced into a held state because it has too many Received: header lines. A value of 1 instructs IMTA to issue a message when this happens. A value of 0 (the default) turns off these messages.

MAX_LOCAL_RECEIVED_LINES (integer)

As IMTA processes a message, it scans any Received: header lines attached to the message looking for references to the official local host name. (Any Received: line that IMTA inserts will contain this name). If the number of Received: lines containing this name exceeds the `MAX_LOCAL_RECEIVED_LINES` value, the message is entered into the IMTA queue in a held state. The default for this value is 10 if no value is specified in the option file. This check blocks certain kinds of message forwarding loops. The message must be manually moved from the held state for processing to continue.

MAX_RECEIVED_LINES (integer)

As IMTA processes a message, it counts the number of Received: header lines in the message's header. If the number of Received: lines exceeds the `MAX_RECEIVED_LINES` value, the message is entered into the IMTA queue in a held state. The default for this value is 50 if no value is specified in the option file. This check blocks certain kinds of message forwarding loops. The message must be manually moved from the held state for processing to continue.

Internal Size Options

These options relate to internal IMTA sizing issues. In general, these options should not be set manually, but should instead be automatically resized when necessary by using IMTA `cnbuild` (UNIX) utility.

ALIAS_HASH_SIZE (integer <= 32,767)

`ALIAS_HASH_SIZE` sets the size of the alias hash table. This in turn is an upper limit on the number of aliases that can be defined in the alias file. The default is 256; the maximum value allowed is 32,767.

ALIAS_MEMBER_SIZE (integer <= 20,000)

`ALIAS_MEMBER_SIZE` controls the size of the index table that contains the list of alias translation value pointers. The total number of addresses on the right sides of all the alias definitions in the alias file cannot exceed this value. The default is 320; the maximum allowed is 20,000.

CHANNEL_TABLE_SIZE (integer <= 32,767)

CHANNEL_TABLE_SIZE controls the size of the channel table. The total number of channels in the configuration file cannot exceed this value. The default is 256; the maximum is 32,767.

CONVERSION_SIZE (integer <= 2000)

CONVERSION_SIZE controls the size of the conversion entry table. The total number of conversion file entries cannot exceed this number. The default is 32.

DOMAIN_HASH_SIZE (integer <= 32,767)

DOMAIN_HASH_SIZE controls the size of the domain rewrite rules hash table. Each rewrite rule in the configuration file consumes one slot in this hash table. The number of rewrite rules cannot exceed this option's value. The default is 512; the maximum number of rewrite rules allowed is 32,767.

HOST_HASH_SIZE (integer <= 32,767)

HOST_HASH_SIZE controls the size of the channel hosts hash table. Each channel host specified on a channel definition in the IMTA configuration file (both official hosts and aliases) consumes one slot in this hash table. The total number of channel hosts cannot exceed the value specified. The default is 512; the maximum value allowed is 32,767.

MAP_NAMES_SIZE (integer > 0)

MAP_NAMES_SIZE specifies the size of the mapping table name table. The total number of mapping table names cannot exceed this number. The default is 32.

STRING_POOL_SIZE (integer <= 10,000,000)

STRING_POOL_SIZE controls the number of character slots allocated to the string pool used to hold rewrite rule templates, alias list members, mapping entries, and so on. A fatal error will occur if the total number of characters consumed by these parts of the configuration files exceeds this limit. The default is 65,000; the maximum allowed value is 10,000,000.

WILD_POOL_SIZE (integer)

`WILD_POOL_SIZE` controls the total number of patterns that may appear throughout mapping tables. A fatal error will occur if the total number of mapping patterns exceeds this limit. The default is 8,000; the maximum allowed value is 200,000.

Debugging Options

These options enable debugging of various IMTA facilities.

DEQUEUE_DEBUG (0 or 1)

`DEQUEUE_DEBUG` specifies whether debugging output from IMTA's dequeue facility QU is produced. If enabled with a value of 1, this output will be produced on all channels that use the QU routines. The default value of 0 disables this output.

POST_DEBUG (0 or 1)

`POST_DEBUG` specifies whether debugging output is produced by IMTA's periodic delivery job. If enabled with a value of 1, this output will be produced in the `post.log` file. The default value of 0 disables this output.

SIMS 4.0 Man Pages

This section describes any changes or updates to the SIMS 4.0 UNIX man pages.

imrestore(1M) (bug 4244175)

An error exists in the "Synopsis" section of the `imrestore`(1M) man page. The correct synopsis is:

```
imrestore [-b blocking_factor] [-t 1 | 2 | 3] [-i] \
[-f device | file | -] [-c continuation_file] [-l config_file] \
[ [-u usernames_file] | userid... | userid@domain...]
```

Only a *usernames_file* is used with the `-u` option. The *userid* and *userid@domain* parameters are not used with the `-u` option.

An addition should also be made to the description for each of the user parameters (*userid*, *userid@domain*, and *usernames_file*) In each of the user parameters, the user or users that are being restored can be listed (either in a file or on the command line). The user can be specified as *userid* or *old_userid=new_userid*. When restoring a user back to the same userid, you only need to specify the *userid*. If restoring or migrating a user to a new *userid*, the equal sign (=) is used to separate the new and the old *userid*s. If a *userid* contains an '=', you need to escape the character with a backslash (\). For example:

```
imrestore bsmith jones=srjones "g=\=chen=gchen"
```

Do not use any blanks around the '=' or within the userids.

Similar syntax should be used if the list is specified in a file (*usernames_file*). However, additional double-quotes should not be placed around any strings specified in a file. The following is an example of a *usernames_file*:

```
bsmith
jones=srjones
g\=chen=gchen
```

This feature is useful when migrating users from SIMS 3.x to SIMS 4.0.

sims.cnf(4) (bug 4246310)

References to the `spmProgramNumber` parameter in the `sims.cnf`(4) man page should be removed. It is not used by the Delegated Management component.

immd_recipient_disposition(3) (bug 4245772)

The "Synopsis" section for the immd_recipient_disposition(3) man pages is incorrect. It reads:

```
#include <imta.h>
int immd_recipient_disposition( immd_t d, const char ‹**rcpt, const
char ‹** orcpt, const char ‹**reason, im_disp_t disp, int flags );
```

The actual function does not accept the flags argument.

imadmin-delete-user(1M) and imadmin-purge-user(1M) (bug 4246863)

The following information should be added to the `imadmin-delete-user`(1M) and `imadmin-purge-user`(1M) man pages:

Clearing the delete flag in the directory does not prevent a user purge. This condition exists during the following times:

- the period between marking the user as deleted and the actual purge
- the period between the time the purge utility detects the delete flag and the time when the purge is performed

imdeluser(1M), imadmin-delete-user(1M), and imadmin-purge-user(1M) (bug 4246867)

The relationship between the `imdeluser`, `imadmin-delete-user`, and `imadmin-purge-user` utilities needs to be described in each of the respective man pages. Add the following description to the `imdeluser`(1M), `imadmin-delete-user`(1M), and `imadmin-purge-user`(1M) man pages:

Similarities and differences exist among the delete user and user purge utilities in SIMS. The `imdeluser` utility removes a specified user or users from the message store. The `imadmin-delete-user` utility only marks a user entry as deleted. In order to remove that user from the message store, you must execute `imadmin-purge-user`. In addition, `imadmin-purge-user` also removes the user's LDAP entry and updates the IMTA cache.