Administration Guide for Scalix Systems

Version 10.0.1

Administration Guide
For Scalix Systems

Published by Scalix Corporation
1400 Fashion Island Blvd., Suite 602
San Mateo, CA 94404-2061
USA

Product Version: 10.0.1

E: 5.11.2006

Notices

The information contained in this document is subject to change without notice.

Scalix Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantibility and fitness for a particular purpose. Scalix Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Unix is used here as a generic term covering all versions of the UNIX operating system. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

Linux is a registered trademark of Linus Torvalds.

Red Hat, and Fedora are registered trademarks of Red Hat Software Inc. rpm is a trademark of Red Hat Software Inc.

SUSE is a registered trademark of Novell Inc.

Java is a registered trademark of Sun Microsystems Inc.

Microsoft, Windows XP, Windows 2000, Windows NT, Exchange, Outlook, Active Directory, and Internet Explorer are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Restricted Rights Legend

Use, duplication, or disclosure is subject to restrictions as set forth in contract subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause 52.227-FAR14.

Contents

Chapter 1

Overview

Scalix now offers two editions of its products: Scalix Community Edition and Scalix Enterprise Edition. The differences between the two are fully detailed in the first section in this chapter.

Following this product overview, the rest of this chapter surveys the major components of the Scalix messaging system and catalogs a list of basic tasks you can perform to begin using the Scalix messaging system.

About Scalix Product Editions

Scalix offers three editions of its powerful email and calendaring platform based on Linux and open systems: Scalix *Enterprise Edition, Small Business Edition* and Scalix *Community Edition*.

Scalix Enterprise Edition is the company's flagship product and is ideal for organizations that demand the full range of functionality in a commercial email and calendaring system. It includes multi-server support, unlimited number of *Standard* users, any number of *Premium* users, the full complement of Scalix advanced capabilities, and a wide variety of technical support options.

Scalix Small Business Edition targets organizations getting started with a commercial version of Scalix that do not have the higher end requirements of Enterprise Edition. It is functionally equivalent to Enterprise Edition except that it allows only single-server installations instead of multi-server, and does not include the capabilities for high availability and multi-instance support.

Scalix Community Edition is the free, single-server, unlimited-use version of the Scalix product and is great for cost-conscious organizations that desire a modern email and calendaring system but do not require advanced groupware and collaboration functionality for their entire user population. It includes unlimited Standard users, twenty-five free Premium users, a subset of Scalix functionality, and fee-based, incident-based technical support.

The following table compares the Scalix product editions in greater detail:

Product Feature	Community Edition	Small Business Edition	Enterprise Edition
User Types			

Standard Users	Free, unlimited	Free, unlimited	Free, unlimited
Premium Users	Maximum 25 premium users (free)	Any number of licensed premium users	Any number of licensed premium users
Core Functionality			
Email & calendaring Server	Single-server	Single-server	Multi-server
Internal user directory	[X]	[X]	[X]
GUI-based installation	[X]	[X]	[X]
GUI & command line administration	[X]	[X]	[X]
Complete documentation	[X]	[X]	[X]
POP/IMAP email client access	Unlimited	Unlimited	Unlimited
Native MS Outlook support (via MAPI)	Premium users only (max 25)	Premium users only	Premium users only
Fully functional AJAX web client (Scalix Web Access)	[X] (group scheduling in calendar for 25 premium users only)	[X] (group scheduling in calendar for all premium users)	[X] (group scheduling in calendar for all premium users)
Native Novell Evolution support	[X] (group scheduling in calendar for 25 premium users only)	X] (group scheduling in calendar for all premium users)	X] (group scheduling in calendar for all premium users)
Public folders	Premium users only (max 25)	Premium users only	Premium users only
High availability	Not available	Not available	[X]
Multiple instances per server	Not available	Not available	[X]
Migration tools	Not available	[X]	[X]
Upgrade To Enterprise Edition	Via license key. Re-installation not required	Via license key. Re-installation not required	Not applicable
Ecosystem Support			
Meta-directory support via LDAP	[X]	[X]	[X]
iCal support	[X]	[X]	[X]
Native Exchange Interoperability (via TNEF)	Not available	[X]	[X]
Active Directory integration with MMC plug-in	Not available	[X]	[X]
Anti-virus	Via flexible 3rd party interface	Via flexible 3rd party interface	Via flexible 3rd party interface

Anti-spam	Via flexible 3rd party interface	Via flexible 3rd party interface	Via flexible 3rd party interface
Archiving	Via flexible 3rd party interface	Via flexible 3rd party interface	Via flexible 3rd party interface
Wireless email & PIM	Email-only via POP/IMAP	Email & PIM via Notify	Email & PIM via Notify
Technical Support			
Community Forum	Free	Free	Free
Knowledgebase, Tech notes	Free	Free	Free
Incident-based Support	Fee-based	Fee-based	Fee-based
Software subscription	Not available	[X]	[X]
Premium 7x24 Support	Not available	[X]	[X]
Cost			
Licensing	Free, unlimited use	Cost based on number of premium users, no cost for Scalix server(s)	Cost based on number of premium users, no cost for Scalix server(s)

Scalix User Types

Scalix users can be defined as *Standard* or *Premium* users, as defined in the following:

Standard Users

Standard users gain access to a subset of Scalix functionality including email, personal calendar and contacts through Scalix Web Access and Novell Evolution as well as email access using POP/IMAP clients. The ability to deploy standard users is ideal for cost-conscious organizations with users who do not have high-end groupware and collaboration requirements. An unlimited number of standard users may be deployed with any Scalix edition for free.

Premium Users

Premium users have access to the full benefits and functionality of the Scalix email and calendaring system. The following Scalix product capabilities are available only to premium users:

- Native MS Outlook support (via MAPI)
- Group scheduling functionality including free/busy lookup in Outlook, Scalix Web Access and Evolution clients
- Access to public folders
- Wireless email and PIM

Any number of licensed premium users may be deployed with Scalix Enterprise Edition. Scalix Community Edition is limited to a maximum of twenty-five (25) free premium users, who enjoy many of the features available to Enterprise Edition premium users.

Flexible, Cost-Effective Email For Everyone

The distinction between standard and premium users provides organizations with the flexibility to cost-effectively provide email for all users. For example, manufacturers and retailers may desire headquarters staff to be designated as premium users as they require advanced groupware capabilities, while less demanding users, such as shop floor or store personnel, would be satisfied as standard users with only email and personal calendaring capabilities. Similarly, educational institutions may decide that faculty and staff are premium users that need advanced collaboration capabilities while students are standard users that just need email and personal calendaring. There is no cost for deploying standard users with either Scalix Community Edition or Scalix Enterprise Edition.

Required Licenses

Scalix *Community Edition, Small Business Edition* and *Enterprise Edition* use the same installer. The main difference is that Small Business Edition and Enterprise Edition require a license key while Community Edition does not. Additionally, if you are a Scalix Community Edition customer, you can only perform the “typical” installation, in which all the Scalix components are stored on a single host computer.

To activate your Scalix system as either a Small Business or Enterprise Edition system, you must enter a license key at a strategic point in the installation process. Please obtain your Scalix license key and have it ready for use before installing Scalix 10.0 or 10.0.1.

You may proceed with the installation without a license key, however, your system will be treated as a Community Edition system and your users as Standard users until the correct license key is entered by means of the *Scalix Administration Console*.

Additionally, you can install Scalix Enterprise Edition onto a single host, or distribute the primary components onto separate hosts—both of which are detailed fully in this guide.

Identifying the Instance Home Directory

Throughout the installation procedure, there are repeated references to the instance’s home directory, known as “~”. The location of this directory varies depending on how you ran your initial setup. For example, if you named the instance when you created it, the home directory becomes /var/opt/om.[name]. But if your instance is unnamed, the home directory becomes /var/opt/scalix.

To determine the home directory for a particular instance, look in /opt/scalix/global/config.

Chapter 2

Configuring Mailnodes and Users

This chapter describes how to configure mailnodes and users on the Scalix system using a command line interface. These tasks can also be completed in the Scalix Administrative Console.

The contents of this chapter includes these topics:

Note	Before you create mailnodes and users on the Scalix system, you must log in to the system as root and set up the path to the Scalix commands (/opt/scalix/bin).

Scalix Mailnodes

A mailnode is a collection of O/R address attributes identifying the location of a Scalix local user. A mailnode is always specified along with a Personal Name address attribute, which gives the identity of the user. A specific mailnode can exist on one only system. Users with the same mailnode cannot be located on different systems.

- Though all Scalix users must be associated with a mailnode, the primary node is the one used by Scalix entities that do not explicitly have an address associated with them. There can be only one Primary mailnode for each local system at any time. For details on the primary mailnode, see the section "Primary Mailnode" on page 6.

Note	An originator recipient address (O/R address) is an attribute list that distinguishes one user, or distribution list, from another and defines the user's point of access to the message handling system or the distribution list's location.

Mailnodes Attributes

A mailnode is composed of the Organizational Unit Name (OU1, OU2, OU3, OU4) O/R Address attribute.

- Organization Name (O)
- Private Domain Name (P)

- Administration Domain Name (A)
- Country Name (C)

While a mailnode can contain all five O/R Address attributes, it might contain only an attribute value for Organizational Unit Name if this is sufficient to identify a user within a local Scalix system. However, when a message with such an address is routed out of the local system, the interface adds the additional attributes in order to fully identify the user.

As for other O/R Address attributes, attributes in a Scalix mailnode can be specified using attribute format, positional format, or mixed format, such as = OU1 (+ OU2 + OU3 + OU4).

```
= OU1 (+ OU2 + OU3 + OU4) + O
= O
= O + P + A + C
= OU1 (+ OU2 + OU3 + OU4) + O + P + A + C
= OU1 (+ OU2 + OU3 + OU4) + P + A + C
```

Remember that mailnodes are specified along with a Personal Name attribute. The Personal Name and mailnode attributes must be specified using printable string characters as shown in the following list.

Character Type	Characters	
Letters (upper case)	A-Z	n/a
Letters (lower case)	a-z	n/a
Digits	0-9	n/a
Space	space	n/a
Punctuation characters	Single quote mark	'
	Parenthesis (left)	(
	Parenthesis (right)	)
	Plus sign	+
	Comma	,
	Hyphen	-
	Period	.
	Forward slash	/
	Colon	:
	Equal sign	=
	Question mark	?

Primary Mailnode

The primary mailnode is the mailnode used when addresses are automatically generated by the Scalix system. Scalix uses the primary mailnode to automatically generate addresses for these entities. For example, when the Service Router routes a message through the Scalix system, it adds a Transport Trace Record to the message. The Service Router uses this record, which contains the host name of the machine and the primary mailnode for that sys-

tem, to detect if a message is caught in a loop (repeating its movement through the delivery path without being delivered).

Primary Mailnode Default and Configuration

There is only one primary mailnode for each local system. Any local mailnode can act as the primary mailnode. By default, the first local mailnode added to the system acts as the primary mailnode.

You can determine which local mailnode is the primary mailnode by using the omshowmn command. This command lists all the mailnodes on the local system, marking the primary mailnode with two asterisks (**) before its name.

You can configure the primary mailnode in the following ways:

- A mailnode can be added to the local system using the omaddmn command. This command also adds to the Routing Table a route for the mailnode to the Local Delivery Service. The first mailnode added to the system by default acts as the primary mailnode.
- One or more local mailnodes can be deleted using the omdelmn command. This command also deletes the local route to the mailnode from the Routing Table. If you delete the current primary mailnode, Scalix designates an arbitrary local mailnode as the primary mailnode.
- Another existing local mailnode can be designated as the primary mailnode using the ommodmn command. The mailnode currently designated as the primary mailnode loses its attribute and becomes a local mailnode.

Creating a Local Mailnode

Before you add users to the Scalix system, you must create at least one mailnode (you can have more than one mailnode per server). The first mailnode you create becomes the "primary" mailnode.

Typically, there are four Organizational Units in a mailnode that can help to identify a by location, job function, or department.

To add a mailnode, enter:

```
omaddmn -m "OU1,OU2"
```

Use the -d option to specify the Internet Domain for the mailnode.

For example, create the mailnode sales,bigco, with Internet Domain sales.bigco.com. When you later create users (e.g., John Doe/sales,bigco)the Internet address of the user will automatically be configured to John.Doe@sales.bigco.com.

Any user or PDL that you later create on this mailnode has their INTERNET-ADDR attribute automatically configured to the Internet Domain you specify for the mailnode.

Use the omaddiam command if you want to associate an Internet Domain with a mailnode so that Internet Addresses are automatically created when you add a user. See the manual page for the omaddiam command for more information.

Note that you can only associate Internet addressing information with mailnodes if automatic Internet address generation is enabled. See “Configuration Options” on page 301 for more information.

For more information about automatically configuring Internet addresses, see “Automatic Generation of Internet Addresses” on page 162.

Mailnode Commands

The following table lists all of the commands associated with mailnodes.

Command	Description
omaddmn	Add a mailnode
omdelmn	Delete one or more mailnodes
ommodmn	Modify a mailnode
omshowmn	List local mailnodes
omaddiam	Associate a mailnode with an Internet Domain
omshowiam	List an Internet Domain associated with a mailnode
ommodiam	Modify an Internet Domain associated with a mailnode
omdeliam	Delete an Internet Domain associated with a mailnode

Scalix Users

The following information describes Scalix user attributes, adding users to the Scalix system, and managing users.

Personal Name Attributes

To identify local mail users, Scalix uses the Personal Name attribute and a mailnode. Personal Name is composed of the following sub-attributes:

- Last Name (S), plus optional
- First Name (G)
- Initials (I)
- Generation qualifier (Q)

Adding a Local User

Before you begin adding users, make sure you add an administrator account (with administration privileges). When you do this, you do not have to login as root to administer Scalix (however, you must have “root” execute many Scalix commands).

1 To add a local user to the Scalix system (with a password of abc123), enter:

```
omaddu -n "John Doe/sales,bigco/" -p abc123
```

This generates the following user record:

```
John Doe/sales,bigco/CN=John Doe
```

Note When you add a new user record, the CN attribute is automatically created by combining the first name and the last name. See the omaddu manual page for more information.

You do not have to configure an Internet Address if you used the -d option while creating the user (as described in “Creating a Local Mailnode” on page 7). However, if you want to add multiple Internet Addresses associated with a user, use the ommodent command to modify the IA field in the directory as follows:

```
ommodent -e G=John/S=Doe -n INTERNET-ADDR=john@sales.bigco.com
```

See “Directory Entries” on page 95 for more information about editing directory entries.

Creating an Error Notification User

You should create a Scalix Postmaster user to which Non-Delivery Reports and any messages addressed to the Error Manager Server are sent.

1 Enter this command:

```
omaddu -n "postmaster/Acme,Boston" -c admin -p error
```

This adds the user (mailbox) with a password of "error".

2 Then enter this command:

```
omconfenu -n "postmaster"
```

This makes the user “postmaster” the Error Notification User.

Configuring the Internet Address for the Error Notification User

To have messages addressed to postmaster@domain to route to the Scalix Error Notification user, you must create postmaster@domain as an Internet-Address for the Error Notification user.

1 Enter the following command:

```
omshowenu
```

This command lists the error notification user.

2 Enter this command:

```
ommodu -o "<enu user>" -n "<enu user>/IA=\"Postmaster\" <postmas-
ter@domain>"
```

3 To prevent postmaster from displaying in the From field of messages sent externally from the Error Notification user (if they already have a Scalix Internet Address), you must add the postmaster as an additional alias:

```
ommodu -o "<enu user>" -n "<enu user>/IA=original IA=postmas-
ter@scalix.com"
```

Configuring User Password Controls

You can configure user password controls using the following variations of the omconfpwd command.

```
omconfpwd [-e expiration_days] [-l minimum_length]
```

The previous format forces users to renew their passwords at regular intervals.

```
omconfpwd [-r repeat_character] [-c options]
```

The previous format forces users to follow certain rules when they set their passwords.

```
omconfpwd [-k illegal_tries] [-u]
```

The previous format locks a user's Scalix account after a specified number of incorrect password entries.

To review the current password configuration, enter omshowpwd.

Listing Active Users

To list active users, use this command:

```
omstat -u service
```

Updating the Directory

Users you add (or delete) to Scalix do not display in Directories until you execute the Client Directory Access (CDA) Server. The CDA builds access tables for Scalix Directories to provide sorted lists of Directory entries. By default, the CDA Server builds the access tables every 1440 minutes (every 24 hours). The omexeccda command forces the immediate processing of a Directory without waiting for the next periodic rebuild of its access tables.

Enter omexeccda to force an immediate build of the directory access tables for the SYSTEM directory (SYSTEM is the name of the default directory).

Managing Users with Active Directory

Scalix Server allows you to use Active Directory to create new mail-enabled objects and synchronize (import) this data to Scalix Server. This enables you to manage users in Active Directory and establish a coexistence environment with Exchange 2000. To do this, you must create a synchronization agreement between Active Directory and the Scalix Server.

You can edit the Synchronization Agreement in the following way:

- interactively using the omldapsync menu
- by launching the VI editor from the menu
- by editing the sync.cfg file in the relevant Synchronization Agreement directory with a Linux editor such as EMACS.

Scalix Corporation recommends that you configure the Synchronization Agreement interactively if you are configuring the agreement for the first time. This allows you to become

familiar with the mapping file while the omldapsync command prompts you for specific information.

See the Scalix Migration Guide for more information about the omldapsync command.

Caution	Use caution when editing the mapping table portion of the Synchronization Agreement. The mapping table is a powerful and flexible component which allows you to determine the data that is imported and exported, and how it appears in the respective target environments. Entering data incorrectly into the mapping table can cause directory replication "storms" and entries displaying in unexpected formats. Scalix Corporation recommends that you execute the directory synchronization process in a test environment before synchronizing directories in a production environment.

To set up an Active Directory mail-enabled object and configure the Synchronization Agreement for Exchange 2000, do the following steps:

Caution	The values you enter for the extensionAttributes (for both Exchange and omldapsync) are case-sensitive.

1. On the Active Directory system, create a new mail-enabled object (otherwise known as a user).
2. When you finish creating this user, select the user and choose Action > Properties > Exchange Advanced tab > **Custom Attributes**.

   When the Custom Attributes window appears, double-click **extensionAttribute10**.
3. Enter `true` and click **OK**.
4. Double-click **extensionAttribute11**.
5. Enter the name of the Scalix Server Mailnode to which you want to export the user and click **OK**.
6. To execute omldapsync in interactive mode, enter the following command:

   ```
   omldapsync -i syncid
   ```

   The omldapsync menu appears.
7. At the prompt, enter `1`.

   The omldapsync command determines that this is the initial directory synchronization and creates the subdirectory for the Synchronization Agreement and the sync.cfg file.

   A series of interactive prompts appears, starting with the following.

   ```
   INPUT: Select sync agreement type to create (00):
   ```
8. Enter 12 to specify that you are synchronizing with an Exchange 2000 server.

   This text appears:

   ```
   2004-04-19 16:18:16 INFO: create file ad_sync/sync.cfg
   ...
   ############################################################
   # NOTE: Some remote LDAP attributes are reserved by Scalix.
   ```

```
# If these have already been used, change the defaults now.
# Otherwise, accept the defaults presented inside brackets.
```

9 A prompt appears:

```
INPUT: value for EX_SCALIX_MAILBOX (extensionAttribute10):
```

10 Press Enter to accept the default (10), or specify an extension attribute, as shown here:

```
extensionAttribute##
```

(Replace "##" with your preferred attribute.)

The following prompt appears:

```
INPUT: value for EX_SCALIX_MAILNODE (extensionAttribute11):
```

11 Press Enter to accept the default, or specify the extension attribute for EX_SCALIX_MAILNODE (for example, extensionAttribute#).

The following information appears:

```
INPUT: value for EX_SCALIX_EXTAUTH (extensionAttribute12):
```

12 Press Enter to accept the default (12), or specify the extension attribute.

The following information appears:

```
INPUT: value for EX_SCALIX_ADMIN (extensionAttribute13):
```

13 Press Enter to accept the default (13), or specify the extension attribute.

The following information appears:

```
INPUT: value for EX_SCALIX_MBOXADMIN (extensionAttribute14):
```

14 Press Enter to accept the default, or specify the extension attribute for EX_SCALIX_MBOXADMIN (for example, extensionAttribute#).

15 At the Edit config file now y/n (n) prompt, enter y.

16 At the Use vi to edit y/n (n) prompt, enter n.

After you are familiar with the Synchronization Agreement process and the VI editor, you can enter y at the Use vi to edit y/n (n): prompt during future configuration changes. For the initial configuration, enter n so that the omldapsync command can step you through configuration of the sync.cfg file.

17 See the *Scalix Migration Guide* to continue the synchronization agreement process.

Importing from Exchange 2000 for the First Time

Use the following command to import data form Exchange 2000:

```
omldapsync -l agreement name
```

All qualified entries in Exchange 2000 are downloaded and imported into the Scalix Server. This command imports entries that are Internet User and Group (PDL) entries representing either:

- mail-enabled mailboxes (Exchange users)

- mail-enabled users (other types of users with e-mail)
- mail-enabled contacts (any entry with e-mail)

Alert	For complete, up-to-date information about using omldapsync in migrating existing user data from an Exchange server to Scalix, please see the separate guide, *Migrating a Mail System to Scalix* (the PDF book "migration_guide.pdf")

Importing Changes from Exchange 2000

Use the following command to import changes (after the initial import) from Exchange 2000:

```
omldapsync -u agreement name
```

All modified and qualified entries in Exchange 2000 are imported into Scalix Server. For example, to add a Scalix Server user:

1. Use Active Directory to create a mail-enabled user, including Scalix Server-designated fields for the mailbox, mailnode, and the host. Also, set the Display Name, Phone Number, and other applicable information.
2. Use the omldapsync -u command to update Scalix Server. The omldapsync -u command creates the new user and the user mailbox in Scalix Server, including mailnode and host information.

Note	The mailnode must already exist. You cannot create new mailnodes in Active Directory.

3. You can update Scalix Server with changes you made to one user in Active Directory by entering:

   ```
   omldapsync -u username -I
   ```

   The process is similar for creating PDLs and Internet users, However, routing information for remote mailnodes must already exist. You cannot create new routing information in Active Directory.

Note	Use a system scheduler such as cron to update directories on a regular (hourly, daily, weekly, etc.) basis.

Managing Errors

During any update, omldapsync creates a search results file (search.curr). This file is used to determine the changes that occurred before the last execution. If the import from Exchange 2000 is successful, search.curr is renamed and replaces the previous copy of the file (search.last). However, if an error occurs during the import, the search.curr file is kept so that the error information is retained. You can then use this information to correct the error(s). omldapsync keeps the search.curr file until no errors occur during the import, or until you use the -A option to accept (ignore) the error condition.

For example:

```
omldapsync -u syncid -I -A
```

Managing Users with OpenLDAP

Similar to Active Directory, you can import of meta-directory information from an OpenLDAP Directory into Scalix. This is allows you to manage Scalix users from the remote system, including mailbox creation and group maintenance tasks. You must specify and reserve a set of Scalix-specific attributes on the remote system (for example, mailbox, mailnode, authentication, and administration attributes). These should be new remote directory attributes created by extending the schema, as described in this schema file:

```
~/sys/ldapsync13.schema
```

You can create the Synchronization Agreement between OpenLDAP and the Scalix Server in any of the following ways:

- interactively using the omldapsync menu
- by launching the VI editor from the menu
- by editing the sync.cfg file in the relevant Synchronization Agreement directory with a Linux editor such as EMACS.

Scalix Corporation recommends that you configure the Synchronization Agreement interactively if you are configuring the agreement for the first time. This allows you to become familiar with the mapping file while the omldapsync command prompts you for specific information.

For more information about using Scalix with OpenLDAP, see the omldapsync man page.

Managing Users with LDAP-compatible Directories

You can import meta-directory information from a LDAP-compatible customized directory to Scalix, and perform various mailbox and administrative tasks. See the “Customized Directory" section in the omldapsync man page for more information.

Chapter 3

Services, Queues, and Daemons

This chapter describes the following Scalix components:

Services

Services are processes that you can enable and disable (using the omon and omoff commands) without adversely affecting the operation of the Scalix system. Each service involved in message management has an associated queue which provides the input that drives the process.

Interfaces to Scalix

Interfaces are used to pass messages from one system to another in a way that enables the messages to reach the destination system in their original form. There are two general types of interface to Scalix:

- The Client interface, which provides the link between the Scalix Server and its clients.
- The Transport interface, which provides the link between Scalix Servers as well as between Scalix Servers and other messaging systems.

The following table lists the interfaces to Scalix.

Interface	Description
Client, including the UAL Local and UAL Remote protocols	Provides the link between Scalix and its clients. Client connections to Scalix use standard networking protocols or serial connections.
Sendmail	Transports messages from one Scalix system to another.

Gateways to Other Messaging Systems

Gateways are the crossing points between Scalix and other messaging systems. A gateway converts a message between the format used by Scalix and the format used by the other system.

Gateways enable Scalix to be used as a backbone messaging system linking incompatible proprietary messaging systems, which in turn produces a standards-based message handling environment.

The gateways that are included with Scalix are described in the following table.

Gateway	Description
Internet Mail Gateway	Handles messaging between Scalix and any internet mail system that uses the Sendmail routing process. It can be configured to use MIME and TNEFmethods of encoding outgoing messages and decoding incoming messages. Note: TNEF routing is available only in Scalix Enterprise Edition. For more information, see "About Scalix Product Editions".

Servers for Internal Mail Operations

There are a number of servers that perform a range of mail operations within Scalix. The servers for Scalix operations are listed in the following table.

Server	Description
Dump	Copies messages before or after they have passed through the Service Router.
Public Folder	Synchronizes items on Public Folders across the network and attaches messages to Public Folders. Note: Public folders can be accessed only by Premium users. For more information, see "About Scalix Product Editions".
Client Directory Access	Used to generate sorted lists of Directory entries for the Outlook client.
Deferred Mail Manager	Handles any messages whose delivery is to be deferred.
Error Manager	Routes messages addressed to it to the locally designated Error Manager.
Local Delivery	Handles the delivery of messages addressed to local users or entities.
omscan	Scans the Message Store and produces reports that are used by the `omscan` command.
Request	Automatically performs preconfigured tasks when users send messages to it.
Search Server	Enables UAL clients to perform background searches of the Message Store for specified container or basic items.
Service Router	Examines all messages coming into Scalix and decides which delivery service they must be passed to—one of the interfaces, gateways, or Local Delivery.
Archiver	Archives all messages traversing a Scalix Server.

List of Services

The following table lists the Scalix services that can you can enable and disable, giving their abbreviations, and identifying the associated processes that are started with the services.

Service	Abbreviations	Process
Internet Mail Gateway	net, unix	unix.out unix.in

Service	Abbreviations	Process
POP3 interface	pop, pop3	pop3.server
UAL local protocol	lci, l-client	ual.local
UAL remote protocol	rci, r-client	advmail.* ual.*
Sendmail Interface	sendmail, sm	xport.out

Service	Abbreviations	Process
Dump Server	dump	xport.out
Public Folder Server	bulletin, bbs	bb.server
Client Directory Access Server	cda	cda.server
Deferred Mail Manager	(The Deferred Mail Manager is started automatically by the Service Router)	defer.manager
Directory Synchronization Server	dirsync	dirsync
Error Manager Server	(The Error Manager Server is enabled automatically by the Local Delivery Service.)	error.manager
Local Delivery Service	local, ld	local.delivery error.manager
omscan Server	omscan, scan	omscan.server
Request Server	request, req	req.server
Search Server	se, search	search.server search.process
Service Router	router, sr	service.router license.server xport.in
Archiver	archiver	service.router

Starting and Stopping Individual Services

To see a list of the Scalix services that are installed on the system, enter the following command:

```
omstat -s
```

This command lists the services, and their status.

To start a service, enter the following command:

```
omon -s service-name
```

service-name is the name of the service. This can be the full name, as listed in the output to the omstat -s command. In this case, insert the service name between quotation marks (“service_name”).

For example, to start the Public Folder Server, you can enter one of the following two commands:

```
omon -s “Public Folder Server”
```

```
omon -s bbs
```

To stop a service, enter the following command:

```
omoff -d0 -s service-name
```

Daemons

A daemon is a background process which spawns a child process to provide a service requested by another Scalix process. The daemon’s child process exists only for the current session.

Some daemon processes start when Scalix is started using the `omrc` command. These daemons can not be shut down while the Scalix system is operating. They are stopped only when the Scalix system is shut down using the `omshut` command.

Other daemon processes can be explicitly started using the `omon -a` command. They are not essential for Scalix to function correctly. These daemons can be shut down using the command `omoff -d delay -a`, where *delay* is the time in seconds to wait before shutting down the process.

For a list of which daemons cannot be shut down while Scalix is running, enter the `omstat -a` command. Those that cannot be shut down are reported as “`non-stop`”.

Daemons do not have associated queues.

List of Daemons

This table lists the Scalix daemons.

Daemon	Description	Abbreviation	Process
Container Access Monitor	Keeps a record of which Message Store container items are opened and by which process.	ctaccess, cam	omctmon
Database Monitor	Periodically checks that open database files are associated with existing processes.	omdbmon	omdbmon
Directory Relay Server	Multiplexes Directory requests to X.500 Directory services, and identifies host servers of mailnodes.	drs	omdrs
Event Notification Monitor	Tracks all notifications registered against any direct reference in the Message Store and ensures that the event information is captured and correctly reported to a UAL client when a `GET NOTIFICATION` command is issued.	nfa	notif.mon
IMAP4 Daemon	Provides an interface to enable IMAP4 clients to connect to Scalix	IMAP4	in.imap41d
Item Delete Daemon	Deletes items that are no longer attached to a Scalix queue.	idd	item.delete
Item Structure Server	Used in conjunction with `omscan` and `omcontain` in single-user restore operations and the repair of the Message Store.	omissdm	omissdm
LDAP Daemon	Provides an interface to enable LDAP clients to store data in and retrieve data from a Scalix Directory. There can be 0 or more of these processes.	omslapd, omslapdeng.	omslapd, omslapdeng.
Mime Browser Controller	Controls a pool of `mime.browse` processes. It assigns one of these processes in response to a request from the UAL or POP3 Server.		
Notification Server	Maintains shared memory, allocating space for each configured user on a machine plus 10 percent of the total amount of allocated space as free slots.	ns, noti	omnsdm

Daemon	Description	Abbreviation	Process
Queue Manager	Controls all Scalix queues, including scheduling and reading queue information from disk at startup.	qmgr, qm	queue.manager
Session Monitor Daemon	Maintains data relating to user sessions that is required by other Scalix processes, such as each user's last sign-on time, number of current sessions, and so on.	smd	omsmdm
Shared Memory Daemon	Creates a set of shared memory segments for holding information in Scalix configuration files.		
UAL Client Interface	Makes connection between Scalix and a UAL client.		
TCP/IP Named Pipes	This daemon spawns the child process `ual.remote`.		advmail.nmpd
TCP/IP Sockets	This daemon spawns the child process `ual.remote`.		`advmail.sckd`

Starting and Stopping Daemons

To see a list of the Scalix daemons that are installed on your system, enter the following command:

```
omstat -a
```

This command lists the daemons, and their status. If a daemon is listed as NON-STOP, it cannot be started or stopped except by starting or stopping Scalix.

To start a daemon, enter the following command:

```
omon -a daemon-name
```

daemon-name is the name of the daemon. This can be the full name, as listed in the output to the omstat -a command. In this case, insert the daemon name between quotation marks ("name"). Alternatively, enter the abbreviated name, as listed in the man page for scalix-server.

For example, to start the LDAP daemon, you can enter one of the following two commands:

```
omon -a "LDAP Daemon"
```

```
omon -a omslapd
```

To stop a daemon (not listed as NON-STOP), enter the following command:

```
omoff -d0 -a daemon-name
```

Queues

Messages and requests are passed to Scalix processes through Scalix queues. A queue has one or more processes reading from it, and one or more processes writing to it.

Queue Structure

Scalix queues are kept in memory, and controlled by a queue manager process. Information is also held on disk in the message pool (the directory ~/msgpool).

When Scalix restarts, the queue manager process reads the information from the message pool, and reschedules the messages in the appropriate queues. If the queue manager process encounters a problem in reading from the message pool, and therefore Scalix cannot start, you can set the general configuration option QM_DONT_READ_MSG_AT_START to TRUE. See “Configuration Options” on page 301 for more information.

A queue control file, ~/msgpool/poolctrl, contains a list of the queues present on the system, and other information about the message pool.

Each queue maintains the following information for each message it holds:

- An item reference for the actual message held in the Scalix Message Store.

  Referencing messages rather than holding them in the queue allows messages to be attached to more than one queue at a time without having to copy the message.

  A "priority" flag that can be one of the following types:

- Urgent
- Normal
- Low

Messages on a queue are processed in order of priority, and for each priority, in the order they were added.

Manipulating Messages on Queues

You can manipulate messages on queues using the omqdump command. This command enables you to list, move, delete, export, import, open, and close messages on queues. You also can use it to view the status of queues as well as read the transaction files and other header information of opened messages.

For full details on how to manipulate messages on queues, see the online manual entry for omqdump.

List of Queues

Each main Scalix process involved in processing messages, such as the Service Router process, has a queue associated with it. The queue provides the input that drives the main process and any auxiliary processes.

The following table lists queues along with the corresponding process and service names. This information is provided as you might need to specify a queue name with some Scalix commands (such as omstat, omshowrt, and omaddrt).

Queue	Process	Service
BB	bb.server	Public Folder Server input.
DMM	defer.manager	Deferred Mail Manager.
DUMP	xport.out	Archive Server.
ERRMGR	error.manager	Error Manager Server input.
ERROR	—	Holds corrupt messages. When a message is moving around a Scalix system, it is put in a transaction file. If the transaction file becomes corrupt, the message is placed on the ERROR queue. Messages on the ERROR queue are examined using `omqdump`. Messages on this queue can be resubmitted to the Service Router using `omresub` when the problem has been corrected.
IDEL	item.delete	Queue Manager. Holds messages that are awaiting deletion.
LOCAL	local.delivery	Local Delivery Service input.
POISON	—	Queue Manager. Holds those messages that have caused a process to die 3 times in a row.
REQ	req.server	Request Server input.
RESOLVE	service.router	The queue itself is not used. Rather, the queue name is used in the Routing Table to tell the Service Router to resolve the address in the system default Directory before again trying to route it (see “Resolve Queue” on page 79).
ROUTER	service.router	Service Router input.
SMERR	—	This queue is used if Sendmail rejects a message passed to it by the Scalix Sendmail Interface. Messages on this queue can be resubmitted to the Service Router using `omresub` when the problem has been corrected.
SMINTFC	xport.out	Sendmail Interface input.
UNIX	unix.out	Internet Mail Gateway input.
ARCHIVE	archiver	Archiver input. Any messages that the Archiver is unable to archive remain in the ARCHERR queue.
ARCHERR	—	If the Archiver cannot archive a message (for example, if a third-party SMTP listener rejects a message), the message is placed on this queue.

Service, Queue, and Daemon Command Summary

The following table lists all of the commands associated with services, queues, and daemons.

Command	Description
omisoff	Verify that Scalix services are off
omoff	Stop one or more services
omon	Start one or more services
omrc	Start Scalix
omreset	Reset status of services or remove Scalix
omresub	Resubmit messages
omresubdmp	Resubmit messages processed by the Archive Server
omsetsvc	Display the status of a service in detail; configure auxiliary processes
omshut	Stop Scalix
omstat	List Scalix daemons

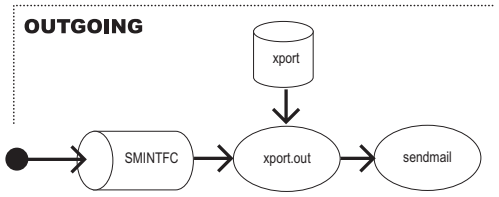
Chapter 4

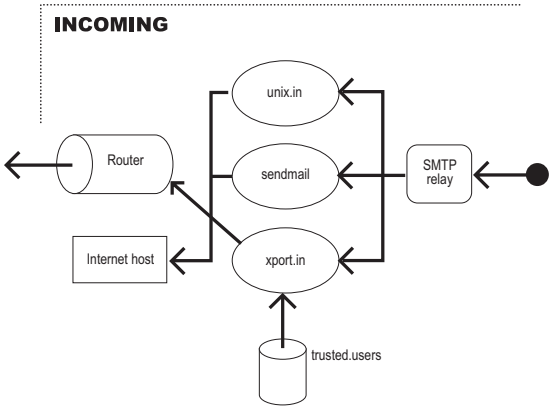
Scalix Interfaces and Gateways

This chapter describes the following Scalix interfaces:

About the Sendmail Interface

The Sendmail Interface passes messages between Scalix systems using the Sendmail transport. The following chart illustrates how the Sendmail Interface manages outgoing and incoming messages.





The following table summarizes key Sendmail Interface information.

Item	Name
Service name	smintfc
Queue name	SMINTFC
Incoming process	xport.in
Outgoing process	xport.out

Outgoing Sendmail Interface

The following steps describe how the Outgoing Sendmail Interface processes outgoing messages:

1. The /opt/scalix/bin/xport.out process receives messages from its input queue SMINTFC.

   The xport.out process encodes and serializes a message. That is, the various files that make up a message are encoded, so that there are no 8-bit characters or control characters, and then placed in a single file.

2. If the xport.out process is unable to encode and serialize a message, the message is attached to the ERROR queue.

3. For each destination system specified by the Sendmail routing information in the active distribution list of a message, xport.out passes to Sendmail one copy of the message.

4. The xport.out process passes a message to Sendmail using Sendmail's standard input.

5. On a heavily loaded system, Scalix can pass messages to Sendmail faster than Sendmail is able to transmit them. When this happens, messages can seem to disappear. They are in fact in the Sendmail queue directory waiting for Sendmail to be allocated sufficient system resources to process them.

Caution Do not modify the xport.out configuration file ~/sys/xport.mappers/XPORT

Incoming Sendmail Interface

The following steps describe how incoming messages are processed:

1. The SMTP Relay receives a message. If the message is destined for the Internet Mail Gateway, it is relayed to unix.in. (If the Internet Mail Gateway is not running for any reason, the message is passed to Sendmail.) If it is destined for the local Scalix system, it is relayed to xport.in. Otherwise it is passed to Sendmail.

   If appropriately configured, the SMTP Relay can perform authentication, to enhance security, help prevent spamming, and provide billing information. See "Virus and Spam Protection" on page 205 for more information.

2. The /opt/scalix/bin/xport.in process receives messages from the SMTP Relay.

The SMTP Relay passes a message to the standard input of xport.in.

3 The xport.in process decodes and then deserializes messages into transaction files and content files, which are then attached to the Service Router queue ROUTER.

- If the Service Router is disabled when the xport.in process receives a message, the message is placed in the ~/xport.hold directory. Messages in this directory are automatically processed by xport.in after the Service Router has been restarted.
- If the xport.in process is unable to deserialize a message, the message is placed in the ~/xport.errs directory, and an ERROR is logged in the Event Log. (The xport.errs directory is not created until it is required.)
- If the xport.in process finds that a message has been returned by a remote system, that is that the original message is enclosed in another message, it is placed on the SMERR queue instead of the ROUTER queue.

To monitor who is sending messages into Scalix, the xport.in process checks the Linux user ID of the process invoking it (the xport.in process is invoked for each message). If the ID is not listed in the ~/sys/xport.mappers/trusted.users file, the message is rejected, and the rejection is logged in the Event Log.

Sendmail Configuration File

Sendmail is used by both the Scalix Sendmail Interface (for Scalix-to-Scalix message transport) and the Scalix Internet Mail Gateway.

Messages leaving Scalix through the Sendmail Interface and the Internet Mail Gateway are passed directly to the Sendmail process. No special configuration of Sendmail is required.

For incoming messages, the SMTP Relay checks the destination address of the message, and passes it to Sendmail, the Incoming Internet Mail Gateway (unix.in), or xport.in, as appropriate. However, if the SMTP Relay is not functioning for any reason, the message can be passed straight to the Sendmail Interface, which can need to route it into Scalix.

It is this process that must be configured in the Sendmail configuration file (/etc/sendmail.cf).

Normally, the Scalix installation process adds the appropriate configuration lines to the Sendmail configuration file, if it finds Sendmail on the system.

If you add Sendmail to your system after installing Scalix, you must ensure that it is configured appropriately for Scalix, as follows:

1 Back up your old Sendmail configuration file (/etc/sendmail.cf).

2 Copy your new Sendmail configuration file to /etc/sendmail.cf.

3 Run the omsendin script.

Changes Made to the Sendmail Configuration File

The following changes are made to the Sendmail configuration file by the Scalix installation script.

- Two new mailers are defined:

scalix - for the Internet Mail Gateway (the Scalix unix.in process).

omxport - for the Scalix transport through the Sendmail Interface (the Scalix xport.in process).

- A mapper process, ommapsmtp, checks each address to see if it should be passed to xport.in or unix.in before processing it for onward relay or local delivery.

Note	The ommapsmtp mapper process is designed for the most general case, and checks all incoming addresses. However, you might be able to enhance the performance of Sendmail by avoiding the use of ommapsmtp. For example, if your Scalix configuration is such that all Scalix addresses are from a small number of domains, you could edit the Sendmail configuration file to use some simple rules instead of ommapsmtp.

SMTP Relay

The SMTP Relay acts to intercept incoming messages and pass them to one of the following processes, depending on their destination address:

- Incoming Internet Mail Gateway (`unix.in`)
- Addresses that contain `/` and end with `@`*`local-domain`*.
- Internet addresses that can be mapped to Scalix addresses using any of the mechanisms described in “Internet Address Mapping” on page 32.
- xport.in

  `scalix@`*`host.local-domain`*
- Sendmail

  All other destination addresses.

Note	You might need to make these changes manually if you upgrade the Linux operating system or Sendmail.

SMTP Relay Configuration File

The SMTP Relay has a configuration file, ~scalix/sys/smtpd.cfg. This allows you to specify aliases for the local domain name, ESMTP extensions supported, maximum hop count, and so on. In addition, you can set options relating to authentication and anti-spamming measures as described in “Virus and Spam Protection” on page 205.

The following table lists SMTP Relay configuration file options (excluding anti-spam and authentication options).

Option	Description
EXTENSIONS	Extensions listed here are advertised by the EHLO reply.
DOMAIN_NAME	Fully Qualified Domain Name of the local host.
LOCAL_NAMES	Local aliases of `DOMAIN_NAME`.

Option	Description
SMTPFILTER	SMTPFILTER has only one state and that's TRUE. So, if you add the following line: SMTPFILTER=TRUE to the ~/sys/smtpd.cfg file, it will cause the SMTP Relay to hand the incoming message off to Sendmail. As a result, inbound messages will be processed through any installed sendmail milters (mail filters) such as Spam Assassin.
DEFAULT_SMTP	Either the name of a program (which must begin with `/`) or a host name and optional port number. The default is `/usr/lib/sendmail -bs`. Note that if mail is being redirected to another machine, that machine must be able to deal correctly with addresses that are nominally local to the originating machine.
MAX_HOP_COUNT	If the number of `Received:` header lines in a message sent to the SMTP Relay exceed this number, the message is rejected by the Relay. The default is 0, indicating that the Relay does not perform loop detection, and loop detection is done only by Sendmail. Any non-positive value is interpreted as infinity.
GREETING	The text in the connection greeting line that appears after the domain name.
TIMEOUT_INITIAL=300 TIMEOUT_HELO=60 TIMEOUT_AUTH=60 TIMEOUT_MAIL=600 TIMEOUT_RCPT=3600 TIMEOUT_DATAINIT=300 TIMEOUT_DATABLOCK=3600 TIMEOUT_DATAFINAL=3600 TIMEOUT_RSET=300 TIMEOUT_QUIT=120 TIMEOUT_MISC=120 TIMEOUT_COMMAND=3600	Timeouts, in seconds, for various states of the SMTP Relay. The values shown are the defaults.
MAX_CLIENTS	Maximum number of clients that can be handled by one `omsmtpd` process. The default is 16. The maximum number of incoming connections is `MAX_CLIENTS` * (`MAX_SUBPROCS + 1`)
MAX_SUBPROCS	Maximum number of `omsmtpd` subprocesses that can be started. The default is 5. The maximum number of incoming connections is `MAX_CLIENTS` * (`MAX_SUBPROCS + 1`)

Internet Mail Gateway

The Internet Mail Gateway passes messages between Scalix and Internet mail systems based on Sendmail.

Three Internet mail message formats are supported:

- RFC 822 format (referred to as SHAR encoding)
- RFC 1521 format (referred to as MIME encoding)

- RFC 1154 format (referred to as UUENCODE encoding)
- MIME Encoded Messages with TNEF Attachment

In addition, the conversion of addresses between Scalix and Internet mail can be customized, as can the conversion of message subjects and body parts.

File name	What it contains
unix.mapper	transport configuration
unixout.str	outgoing body part conversions (default)
sharout.str	outgoing body part conversions (for SHAR encoding)
uxuuout.str	outgoing body part conversions (for UUENCODE)
mimeout.str	outgoing body part conversions (for MIME)
unixin.str	incoming body part conversions (default)
mimein.str	incoming body part conversions (for MIME)
tnefout.str	incoming and outgoing MIME encoded messages with TNEF attachment
unixout.rules	outgoing name conversions
unixin.rules	incoming name conversions
unixmap.out	outgoing address mappings
unixmap.in	incoming address mappings
unixmap.gw	incoming gateway address mappings
unixin.trust	incoming “trusted users” list
mime.types	file code to MIME content-type mappings
mime.cs	character set name mappings

Note TNEF routing is available only in Scalix Enterprise Edition. For more information, see "About Scalix Product Editions".

The following table summarizes key Internet Mail Gateway information.

Item	Name
Service name	net
Queue name	UNIX
Incoming process	xport.in
Outgoing process	xport.out

Configuring the Internet Mail Gateway for Scalix

To configure the Internet Mail Gateway, configure the default mailnodes for incoming messages (see “Configuring Default Mailnodes for the Internet Mail Gateway” on page 31). You can also:

- Add further routes to the gateway. See “Adding a Route to the Local Internet Mail Gateway” on page 31.
- Configure automatic Internet address generation and mapping. See “Configuring Automatic Internet Address Mapping” on page 33.
- Limit the number of addresses that can be output in the ARPA header. See the online manual entry for the command omconfux.
- Change the default body part conversions specified in the steering files: unixout.str,sharout.str, uxuuout.str,mimeout.str, tnefout.strunixin.str, and mimein.str. See “Body Part Conversion” on page 48.
- Add to the list of mappings between MIME content-types and Scalix file codes in the file mime.types. See “MIME Encoded Messages” on page 44.
- Add to the list of mappings between character set names used in MIME and character set names used in Scalix in the file mime.cs. See “MIME Encoded Messages” on page 44.
- Change the default format used to represent O/R Addresses in outgoing messages. See the online manual entry for the command omconfux.
- Change the default encoding method for special characters in outgoing addresses. See the online manual entry for the command omconfux.
- Change the default character (/) used as the primary delimiter for O/R Addresses when they are represented in Internet mail. See the online manual entry for the command omconfux.
- Change the default name conversions specified in the files unixout.rules and unixin.rules. See “Name Conversion” on page 41.
- Configure explicit address mappings in the files unixmap.out, unixmap.in, and unixmap.gw. See “Format of Address Mapping Configuration Files” on page 35.

After completing the configuration, start the Internet Mail Gateway service.

Configuring Default Mailnodes for the Internet Mail Gateway

You can configure default mailnodes for the Internet Mail Gateway using the command:

```
omconfux [-m mime_mailnode] [-u uuencode_mailnode] [-s shar_mailnode] [-t
tnef_mailnode]
```

Note	TNEF routing is available only in Scalix Enterprise Edition. For more information, see "About Scalix Product Editions".

Adding a Route to the Local Internet Mail Gateway

You can add a route to the local Internet Mail Gateway using the command:

```
omaddrt -m OR_address -q UNIX -i {mime | tnef}
```

Internet Address Mapping

An Internet mail message that passes into Scalix must have its Internet mail addresses mapped to Scalix addresses. Similarly, a Scalix message that is sent to the Internet must have its Scalix addresses mapped to Internet addresses.These mappings take place at the IMAP and POP3 interfaces, in addition to the Internet Mail Gateway.

Overview

How does a message that is originated in Scalix get delivered to an Internet address? The simplest case is when the message contains a valid Internet address. However, if the message does not contain a valid Internet address, Scalix requires a method of determining the address from the Scalix address.

There are number of methods by which Scalix can do this. Name Mapping, involving Directory lookup, is the recommended method for most cases, and is described in “Name Mapping” on page 33.

Similarly, for messages from the Internet to Scalix, Name Mapping is the recommended method by which the destination Scalix address is determined.

By default, Scalix automatically creates the INTERNET-ADDR attribute in Directory entries for Scalix users.

Outgoing Messages (unix.out)

Messages going from Scalix to the Internet to a recipient whose address appears in the message uses this address. If there is no Internet address in the message but an address is configured in the DDA fields (whether in the message itself or in the entry retrieved from the Directory), this DDA address information is used rather than the INTERNET-ADDR attribute value configured for that recipient. If no DDA information is available, name mapping is used (if enabled). If an INTERNET-ADDR value is found, this information is used and no other mapping or conversion is attempted for this name.

Incoming Messages (unix.in)

Messages coming in to Scalix through the Internet Mail Gateway use the INTERNET-ADDR attribute value, if present, to map the originator, recipient, and reply-to names and no other mapping or conversion is attempted for this name.

Automatic Internet Address Mapping

Automatic address mapping enables the Internet Mail Gateway to automatically map between Scalix addresses and Internet addresses as messages pass between Scalix and the Internet.

When automatic Internet address mapping is enabled, addresses are determined as follows:

- Outgoing messages

  The Internet address for outgoing messages is determined as described in the section “Outgoing Messages (unix.out)” on page 32.

In this case, Name Mapping is enabled. Since Directory entries are likely to have INTERNET-ADDR attributes defined, due to automatic Internet address generation, further address mapping measures are unlikely to be required.

- Incoming messages

  The Internet address for incoming messages is determined as described in the section “Incoming Messages (unix.in)” on page 32. Again, since Directory entries are likely to have INTERNET-ADDR attributes defined, due to automatic Internet address generation, further address mapping measures are unlikely to be required.

For minimal configuration at the Internet Mail Gateway, all Scalix users and PDLs should have an associated Internet address. Internet addresses can be generated automatically. This ensures that Scalix users and PDLs in the System Directory have associated IA attributes.

The **“Name Mapping”** section below describes the complete process of how Internet and Scalix addresses are mapped, including what happens if automatic Internet address mapping is not enabled. If automatic Internet address mapping is enabled, much of the information in **“Name Mapping”** is not relevant.

Configuring Automatic Internet Address Mapping

Automatic Internet address mapping is enabled when the configuration option INET_USE_AUTO_IAM is set to TRUE (default setting). This also enables the automatic generation of Internet addresses.

To enable Internet address mapping without enabling the automatic generation of Internet addresses, do not configure Internet domains for the mailnodes you create.

Automatic Internet address mapping is disabled when the configuration option INET_USE_AUTO_IAM is set to FALSE.

Name Mapping

To configure mappings between O/R Addresses and Internet mail addresses, you can use the INTERNET-ADDR (or IA) Directory attribute to configure name mappings for individual users. Internet mail addresses will then appear as simple O/R Addresses within Scalix and, conversely, O/R Addresses will appear as simple Internet mail addresses within Internet mail.

The INTERNET-ADDR Directory attribute is a keyed attribute with the attribute number 167. To configure name mapping for a user, enter a value that contains the user’s name and domain name in the format normally expected by Sendmail, for example:

```
joe@pluto.com
```

Mappings occur only when there is an exact match between the name and address in the message and the Directory entry attribute INTERNET-ADDR.

Note	The INTERNET-ADDR attribute (167) can contain comments. Attribute 168 is an automatically generated probe attribute that consists of the formal Internet address part of attribute 167 (that is, without comments), and this is the attribute that is actually used in name mapping.

For Scalix and non-Scalix users, add the IA attributes to entries in the SYSTEM directory This ensures that the Internet addresses are placed in messages, so avoiding the need for further directory lookups. Alternatively, you can add the IA attributes directly to the name mapping Directory. This enables Directory lookups, but does not place the Internet addresses in the messages.

To ensure that name mapping at the Internet Mail Gateway is enabled, make sure that automatic Internet address mapping is enabled (see “Automatic Internet Address Mapping” on page 32) and that the following options in the general.cfg file are set to TRUE:

```
UXO_NAME_MAPPING

UXI_NAME_MAPPING
```

To disable name mapping but retain automatic Internet address generation, set the above options to FALSE.

The following configurable options are also available:

- UX_NAME_MAPPING_DIR
- UX_NAME_MAPPING_DIR_PASSWD
- UX_NAME_MAPPING_ATTRIB

Once enabled, name mapping is applied to all Internet Mail Gateway messages whether routed through SHAR, UUENCODE, MIME, or TNEF.

Example of Name Mapping

In the following example, two users have the following Scalix Directory entries:

```
S=Wolf/G=Chris/OU1=Scalix/INTERNET-ADDR=chrisw@mars.com
S=Smith/G=Joe/OU1=Internet/INTERNET-ADDR=joe@pluto.com
```

With these name mappings configured, Joe Smith on the Internet can address his message to the Scalix user Chris Wolf using an Internet address of chrisw@mars.com. When it arrives, the sender’s Internet mail address (joe@pluto.com) is mapped, and displayed to the recipient as Joe Smith/internet. The recipient’s Internet mail address (chrisw@mars.com) is mapped and displayed to the Scalix recipient as Chris Wolf/scalix.

When Chris Wolf replies to the message, the opposite mappings occur: the reply is addressed to the Internet mail address of the original sender (joe@pluto.com) and the sender name of Chris Wolf/scalix is mapped to the Internet address configured for Chris Wolf in the INTERNET-ADDR attribute (chrisw@mars.com).

Remember you must have routes set up within Scalix that correspond to configured name mappings. In the example used above, the following Scalix route is required:

```
UNIX internet,*,*,* MIME
```

Address Mapping

Use the files unixmap.out, unixmap.in, and unixmap.gw to configure mappings between mailnodes and Internet mail address domains. Internet mail addresses will then appear as simple O/R Addresses within Scalix and, conversely, O/R Addresses will appear as simple Internet mail addresses within Internet mail.

Note that if any of these files are used, the ability of the gateway to automatically use the default mailnode appropriate to the encoding of the original message will be lost. This means that a reply, from a Scalix user to an Internet mail message, can not use the same encoding (MIME or TNEF) as the original message; the encoding will depend on how the return address has been routed to the gateway.

Note	TNEF routing is available only in Scalix Enterprise Edition. For more information, see "About Scalix Product Editions".

Format of Address Mapping Configuration Files

The address mapping configuration files unixmap.in and unixmap.gw have the following format:

```
domain#mailnode#
```

The address mapping configuration file unixmap.out has the following format:

```
mailnode#domain#
```

Where domain is:

```
subdomain....subdomain.subdomain
```

and mailnode is a series of tag$value pairs, separated by periods (.).

```
tag$value....tag$value.tag$value
```

An example entry in the unixmap.in file is:

```
alpha.beta.scalix.com#OU$ux.OU$local.#
```

Tags and their values must be specified in increasing order of significance. If more than one Organizational Unit Name is specified, then the right-most Organizational Unit Name is the most significant. Valid tags, in increasing order of significance, are:

- OU: Organizational Unit Name
- O: Organization Name
- PRMD: Private Domain Name
- ADMD: Administration Domain Name
- C: Country Name

Note the following:

- Lines beginning with a hash character (#) are regarded as a comment line.
- If a value contains a period (.), it must be prefixed with a backward slash (\).
- Any tag with a value of @ is regarded as not present.
- If the domain is @, it is regarded as not present.

unixmap.out

This file configures the mappings of mailnodes to Internet mail address domains in outgoing messages. The mappings in this file must be the reverse of those in the unixmap.in file. (Personal Names are mapped according to the address encoding specified for the gateway by the omconfux -a command.)

It is used when an address in a Scalix message is not routed through the gateway or is routed through the gateway and contains no Internet mail address in its DDA or in its system default Directory entry.

By default, this file is not installed as part of the standard Scalix system and must be created under ~/sys. When created, it must be readable by anyone.

unixmap.out is a text file. Edit it using any standard text editor. The format of the file is described later in this section.

Remember to route any mailnodes you specify through the gateway if the mailnodes are a result of mapping original Internet mail addresses.

unixmap.in

This file configures the mappings of Internet mail address domains to mailnodes in incoming messages. The mappings in this file must be the reverse of those in the unixmap.out file. (Linux user names are mapped according to the address encoding specified for the gateway by the omconfux -a command.) It is used when an incoming Internet mail message contains an address whose domain matches a domain specified in the file.

By default, this file is not installed as part of the standard Scalix system and must be created under ~/sys. When created, it must be readable by anyone.

unixmap.in is a text file. Edit it using any standard text editor. The format of the file is described later in this section.

Remember to configure Sendmail to pass messages containing the domains you specify to the mailer (scalix) used by the gateway if the domains are a result of mapping original Scalix addresses.

Any attribute or sub-domain remaining in the unixmap.in file after the initial mapping is mapped to the next less significant attribute or sub-domain, providing these less significant attributes or sub-domains contain only the characters A-Z, a-z, 0-9, or hyphen (-). For example, the unixmap.in file contains the following entry:

```
alpha.com#OU$ux.OU$alpha.OU$non-scalix.#
```

This causes the Internet address jmg@uk.alpha.com to be mapped to the Scalix address jmg/non-scalix,alpha,ux,uk.

unixmap.gw

This file configures the mappings of Internet mail address domains to mailnodes in incoming messages. (The O/R Address surname is derived from the Internet mail address using the rules specified in the file unixin.rules.)

It is similar to the unixmap.in file but is used when an Internet mail address, in an incoming message, is not mapped by the unixmap.in file and does not contain embedded within it an O/R Address.

The actual Internet address is held completely in the DDA. For example, the unixmap.gw file contains the following entry:

```
alpha.com#OU$ux.OU$alpha.OU$non-scalix.#
```

This causes the Internet address jmg@uk.alpha.com to be mapped to the O/R address jmg/non-scalix,alpha,ux. However, the real Internet address, to be used for replies, is the value stored in the DDA.

By default, this file is not installed as part of the standard Scalix system and must be created under ~/sys. When created, it must be readable by anyone.

unixmap.gw is a text file. Edit it using any standard text editor. The format of the file is described earlier in this section.

Remember to route any mailnodes you specify through the gateway.

Examples of Address Mapping

Your organization has a design group that uses Internet mail and a sales group that uses Scalix. You want to disguise from the two groups the differences in addressing.

For example, the Scalix user Celeste O'Shea/ny,sales,mis wants to send messages through the gateway to a colleague in the design group with an Internet mail address of Nancy.West@mars.pw.com.

With the following mappings configured and the address pattern ny,hq,design routed to the local Internet Mail Gateway, Celeste can send messages to Nancy using the address Nancy West/ny,hq,design.

```
OU$design.OU$hq.OU$ny#mars.pw.com#unixmap.out
mars.pw.com#OU$design.OU$hq.OU$ny#unixmap.in
```

Similarly, with the following mappings configured, Nancy can send messages to Celeste using the address Celeste.O'Shea@sales.pw.com.

```
sales.pw.com#OU$mis.OU$sales.OU$ny#     unixmap.in
OU$mis.OU$sales.OU$ny#sales.pw.com#     unixmap.out
```

Remember, you also need to configure Sendmail to pass messages with a domain of sales.pw.com to the mailer (scalix) used by the gateway.

To catch all other Internet mail messages that can be sent to your Scalix users, use omconfux to configure the default mailnodes along the lines of:

```
omconfux -m ny,hq,internet-mime-external
```

```
-t ny,hq,internet-tnef,external
```

Then, assuming all your internal Internet mail users are within the domain pw.com, configure the following mappings in the unixmap.gw file:

```
@#OU$internet-mime-internal.OU$hq.OU$ny#
pw.com#OU$internet-mime-internal.OU$hq.OU$ny#
```

Remember, you also need configure a route to the gateway for the address pattern ny,hq,internet-mime-internal. For example:

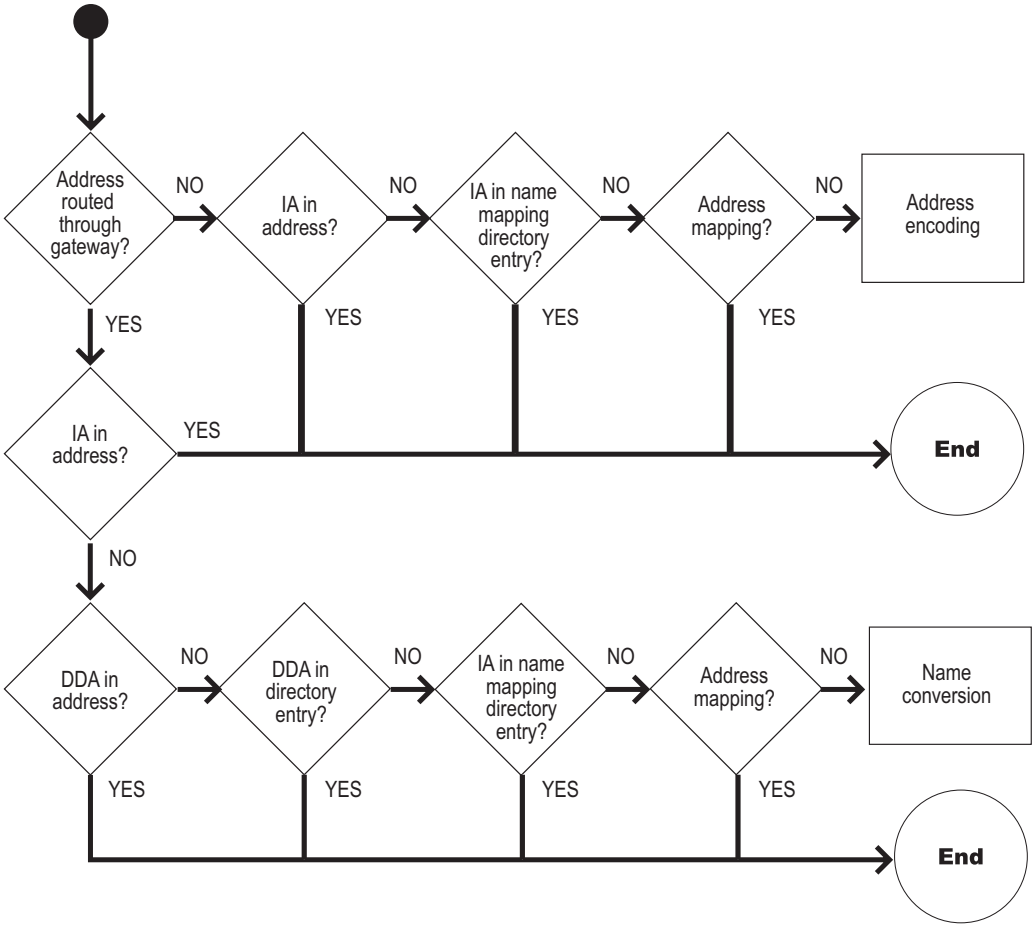
```
omaddrt -m "ny,hq,internet-mime-internal"-q UNIX -i mime
```

If all your internal Internet mail users do not use MIME-aware mailers, you can want to refine the mappings in the unixmap.gw file to distinguish the groups using MIME and standard (RFC 822) Internet mailers.

Outgoing Address Mapping and Name Conversion

The way in which O/R Addresses are converted to Internet mail addresses is outlined here. Some of the rules are configured using the command omconfux (see the online manual entry for details), the remainder of the rules are specified in configuration files (details of which are given in the section “Address Mapping”).

Outgoing Address Mapping and Name Conversion Schema

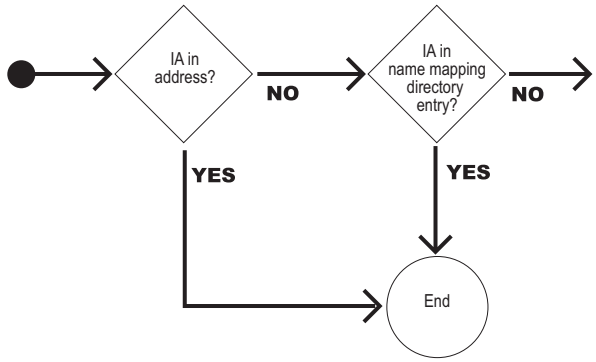


Action	Description
Address routed through gateway	Is the O/R Address routed to the Internet Mail Gateway queue UNIX? Use omshowrt to see a list of address patterns routed to the UNIX queue.
IA in address	Is the INTERNET-ADDR attribute specified in the address?
IA in name mapping directory entry	(Only if name mapping is enabled.) Does the entry for the address in the Directory used for name mapping (normally SYSTEM but can be overridden by the UX_NAME_MAPPING option) have an associated INTERNET-ADDR attribute?

Action	Description
DDA in address	Does the O/R Address have a Domain Defined attribute containing a valid Internet mail address?
DDA in Directory entry	Does the system default Directory entry for the O/R Address have a Domain Defined attribute containing a valid Internet mail address?
Address mapping	Does the O/R Address match a pattern specified in the configuration file unixmap.out? If it does, the corresponding Internet mail domain is used along with a Linux user name derived from the Personal Name. (The format of the user name is specified by the command omconfux -a).
Address encoding	The O/R Address is encoded into a form suitable for Internet mail using the format specified by the command omconfux -a.
Name conversion	The Internet mail address is derived from the surname in the O/R Address using the rules specified in the file unixout.rules.

The previous illustration provides the complete flow. Note that, in most circumstances, this will be simplified to the following:

Outgoing Address Mapping and Name Conversion Flow



The following example shows one method by which O/R Addresses are converted to a form suitable for Internet mail. Default configurations are assumed. The mailnode Internet is routed through the gateway.

```
TO: Joe Smith/Internet/IA=js@abc
TO: nick-at-luna/Internet
FROM: Chris Wolf/ny,hq,mis
```

Joe's address is routed through the local Internet Mail Gateway and contains an INTERNET-ADDR (IA) attribute; therefore this is used as the Internet mail address.

Nick's address is also routed through the local Internet Mail Gateway but does not contain an INTERNET-ADDR attribute. There is no Directory entry for Nick. Therefore the surname in the O/R Address is used to derive an Internet mail address.

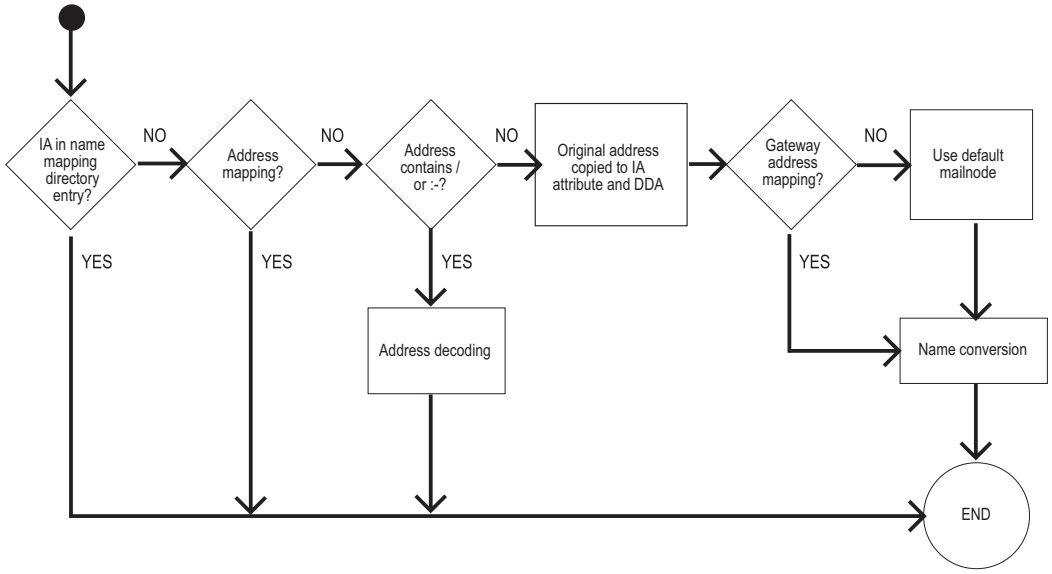
Chris' address is not routed through the local Internet Mail Gateway, therefore the O/R Address is encoded and the local host name added.

```
TO: js@abc
TO: nick@luna
FROM: Wolf_Chris/ny_hq_mis@mars
```

Incoming Address Mapping and Name Conversion

The way in which Internet mail addresses are converted to O/R Addresses is outlined here. Some of the rules are configured using the command omconfux (see the online manual entry for details), the remainder of the rules are specified in configuration files (details of which are given in the section “Address Mapping” on page 34).

Internet Mail Address Conversion to O/R Addresses



Action	Description
IA in name mapping directory entry	(Only if name mapping is enabled.) Does the entry for the address in the Directory used for name mapping (normally SYSTEM but can be overridden by the UX_NAME_MAPPING option) have an associated INTERNET-ADDR attribute?
Address mapping	Does the Internet mail address match a pattern specified in the configuration file unixmap.in? If it does, the corresponding mailnode is used.
Name conversion	The O/R Address surname is derived from the Internet mail address using the rules specified in the file unixin.rules.
Address contains / or;	Does the Internet mail address contain a forward slash (/) or semicolon (;)? If it does, the gateway assumes the address is an encoded O/R Address.

Action	Description
Address decoding	The Internet mail address is decoded into a standard O/R Address. The Internet mail address formats that the gateway can decode are described in the online manual entry for the command omconfux.
Original address copied to IA attribute and DDA	The full Internet mail address, including comments, is copied to a INTERNET-ADDR attribute. The Internet mail address, without comments, is also copied to a Domain Defined Attribute (DDA) in the new O/R Address.
Gateway address mapping	Does the Internet mail address match a pattern specified in the configuration file unixmap.gw? If it does, the corresponding mailnode is used.
Use default mailnode	One of the default mailnodes, configured for the gateway, is used. The selection of the mailnode is determined by the format of the incoming Internet mail message (see later in this section for details). If the appropriate mailnode is not configured, the MIME mailnode is used.

The following example shows how Internet mail addresses, one of which originated from a Scalix system, are converted to a form suitable for Scalix. The Internet mail message contains a MIME header token. The default mailnode for MIME encoded messages is ny,hq,internet-mime.

```
TO: fred@bigco
TO: Wolf_Chris/ny_hq_mis@mars
FROM: js@abc
```

- The first address appears in the SYSTEM Directory, which in this case is the Directory that is used for name mapping. The corresponding Scalix address is added.
- The second address contains a forward slash character (/); therefore, the address is decoded.
- The third address does not contain a forward slash (/) or a semicolon (;); therefore, the address is copied into the INTERNET-ADDR attribute and a Domain Defined Attribute (DDA) and the default mailnode for the gateway is used along with a surname derived from the Internet mail address.

```
TO: Fred Jones/sales,bigco
TO: Chris Wolf/ny,hq,mis
FROM: js/ny,hq,internet-mime/IA=js@abc (js@abc)
```

Name Conversion

Use the file unixout.rules to configure how X.400 surnames are converted to Internet mail addresses. Similarly, use the file unixin.rules to configure how Internet mail addresses are converted to X.400 surnames. These files, together with the default mailnodes configured for the gateway, ensure that messages passing between Scalix and Internet mail always have addresses that are valid within each mail system.

unixout.rules and unixin.rules are text files. Edit them using any standard text editor. The pattern matching rules each uses are documented in the header of the files.

unixout.rules

This file, held under ~/sys, configures the conversion of X.400 surnames to Internet mail addresses in outgoing messages. It is used when an address in a Scalix message is routed through the gateway and contains no Internet mail address in its DDA or in its system default Directory entry, and it does not match any pattern specified in the address mapping configuration file unixmap.out.

The default conversion is from an X.400 surname of the form name-at-subdomain'subdomain... to an Internet mail address of name@subdomain.subdomain....

unixin.rules

This file, held under ~/sys, configures the conversion of Internet mail addresses to X.400 surnames in incoming messages. It is used when an address in an Internet mail message does not match any pattern specified in the address mapping configuration file unixmap.in, and does not contain embedded within it an O/R Address.

The default conversion is from an Internet mail address of the form subdomain!subdomain!name@subdomain.subdomain... to an X.400 surname of name.

Outgoing Internet Mail Gateway

Messages with addresses routed to the Internet Mail Gateway are attached by the Service Router to the input queue of the Internet Mail Gateway (UNIX). The outgoing gateway process /opt/scalix/bin/unix.out takes messages from the queue UNIX and converts them to a form suitable for Internet mail:

- The message header is mapped into an ARPA header.

  See "Message Heading Mappings at the Internet Mail Gateway" on page 50 for details.

- O/R Addresses are converted to Internet mail addresses.

  The exact process depends on the configuration of the gateway and the address being converted.

- The body of the message is concatenated.

  The resultant format of the message is based on one of the following:

- RFC 1521 (MIME encoding).
- MIME encoding, as above, but with an additional WINMAIL.DAT attachment to contain MAPI message properties.
- RFC 822 with an extension for binary parts (SHAR encoding).
- RFC 1154 (UUENCODE encoding).

The format is determined by the route information configured in the Routing Table. See the section "Internet Mail Message Formats" on page 43 for details.

In addition, body parts can be converted. See the section "Body Part Conversion" on page 48 for details.

As the unix.out process converts the message, it passes its output to Sendmail using the Simple Mail Transfer Protocol (SMTP).

If the message contains X.400 security attributes (the transaction file contains either “Message Security Label” or “Security Category” records), the message is returned to the originator.

If the message contains more than 100 Internet mail recipients, then the message is passed to Sendmail once for each set of 100 recipients.

Incoming Internet Mail Gateway

Sendmail checks the recipient addresses in a message to determine whether they are for the local Scalix Internet Mail Gateway. If the address does not contain a %, !, or @ character (that is, the message is local) but does contain a / or; character, the message is passed to the mailer scalix (defined in the Sendmail configuration file as the process /opt/scalix/sys/unix.in).

If Sendmail tries to pass a message to the unix.in process and the gateway is not enabled, an error is returned to Sendmail. Sendmail retains the message and tries again later. If after a about three days (this is configured in Sendmail) Sendmail still fails to deliver the message, it returns a Non-Delivery Report to the originator.

To monitor who is sending messages into Scalix through the Internet Mail Gateway, the unix.in process checks the Linux user ID of the process invoking it (the unix.in process is invoked for each message). If the ID is not listed in the ~/sys/unixin.trust file, the originator’s name in the message is changed to the Linux user ID of the process invoking unix.in.

The unix.in process converts each message to a form suitable for Scalix:

- The ARPA header is mapped into a Scalix header.
- Internet mail addresses are converted to O/R Addresses.

  The exact process depends on the configuration of the gateway and the address being converted.
- If necessary, the message is decoded and its body parts identified and converted.

  See “Body Part Conversion” on page 48 for details.

The unix.in process then attaches the message to the Service Router queue ROUTER.

Internet Mail Message Formats

Four Internet mail message formats are supported. They are automatically recognized in incoming messages and can be generated for outgoing messages. They are:

- RFC 1521, here referred to as MIME encoding.
- MIME encoding, as above, but with an additional WINMAIL.DAT attachment to contain MAPI message properties in TNEF format.
- RFC 1154, here referred to as UUENCODE encoding.
- RFC 822, the classic Internet mail format, here referred to as SHAR encoding.

Note	TNEF routing is available only in Scalix Enterprise Edition. For more information, see "About Scalix Product Editions".

If the option UXI_TREAT_AS_MIME_SUBJECT, in the General Configuration File, is set to T or Y, incoming messages from the Internet Mail Gateway's UUENCODE or SHAR route that have MIME-conforming subjects, have their subjects decoded as if they came via the MIME route. For more information, see "Configuration Options" on page 301.

The following sections describe how each of these formats are handled in incoming and outgoing messages.

MIME Encoded Messages

Incoming RFC 1521 format Internet mail messages (MIME encoding) are recognized by the Internet Mail Gateway and automatically converted to a standard Scalix header and one or more body parts. All five MIME content-transfer-encodings (7-bit, 8-bit, binary, quoted-printable, and base64) are supported. Parameters associated with a content-type are preserved. If a content-description exists for a body part, it is taken as the body part subject.

The MIME content-type text/plain is mapped to a textual body part (file code 1167) of the same character set. The MIME content-types text/enriched, image, video, and application and their subtypes are mapped to body parts with the file codes specified in the file ~/sys/mime.types. If a mapping is not present in the mime.types file, the body part is marked as binary (file code 0) except for content-types of multipart and message which are treated as special cases.

The mime.types file is a text file that can be edited with any standard text editor. The format of the file is documented in the header of the file. Further mappings between file codes and content-types can be specified in this file.

The content-type multipart has a number of subtypes which are interpreted as follows:

Subtype	Meaning
Alternative	All the enclosed body parts are preserved and all except the last are marked as hidden, alternative parts.
Parallel	All the enclosed body parts are placed in a Scalix package and all except the last are marked as parallel parts.
Digest	Each part is mapped to a nested Scalix message, the whole enclosed in a Scalix package.
Mixed	Each part is placed in the Scalix message.

Any other multipart subtypes are treated as multipart/mixed.

If a multipart body part is enclosed within another body part other than message/rfc822, then all the body parts within the multipart body part are enclosed in a Scalix package.

The content-type message has a number of subtypes which are interpreted as follows:

Subtype	Meaning
rfc822	This is mapped to a Scalix message containing a distribution list and body parts. The ARPA header is preserved and marked as hidden.

Subtype	Meaning
External-body	The referenced body-part is not fetched but the reference to it is preserved although not necessarily displayed by a client.
Partial	This subtype is not supported. Content-types of message/partial are placed in Scalix messages as binary attachments.

Any other message subtypes, if not explicitly mapped in the mime.types file, are mapped into binary body parts.

For outgoing messages, the message format is determined by the route information for that route. If it is MIME, the gateway generates a Internet mail message that conforms to RFC 1521. The file codes of Scalix body parts are mapped to MIME content-types as specified in the ~/sys/mime.types file.

If a mapping is not present in the mime.types file, the body part is mapped to the content-type application/x-scalix-file_code[-char_set]. If MIME parameters are present, these are output. Textual body parts use the content-transfer-encoding quoted-printable. All other body parts are encoded using base64. Where a non-textual body part can be converted to text (as specified in the file unixout.str), the textual representation is output as text/plain within a content-type of multipart/alternative along with the original. Body parts with a file code of 1168 (ARPA Header) are not output.

The mime.types file is a text file that can be edited with any standard text editor. The format of the file is documented in the header of the file. Further mappings between file codes and content-types can be specified in this file.

Scalix message structures (nested messages, packages, and so on) are preserved as follows:

Content-Type	Meaning
multipart/alternative	This is generated for a sequence of body parts marked as alternatives. In addition, any conversion to text of a non-textual body part is also output as multipart/alternative.
multipart/parallel	This is generated for a sequence of body parts marked as parallel.
multipart/mixed	This is generated whenever there is more than one body part in a message.
message/rfc822	This is generated for any nested message, reply, and so on.

The following options in the General Configuration File (~/sys/general.cfg) can be of use in dealing with MIME-encoded messages:

BRW_MIME_FNAME_ENCODING
UX_MIME_SUBJECT_CHARSET
UXI_TREAT_AS_MIME_SUBJECT
UXO_MIME_FNAME_ENCODING
UXO_MIME_OMIT_DEF_CTENC_HDR
UXO_MIME_SUBJ_BENC_NONASCII
UXO_MIME_SUBJ_NO_SPACE_SEPS
UXO_MIME_SUBJECT_ENCODING
UXO_MIME_SUBJECT_FOLDING
UXO_MIME_TEXTFILE_ENCODING
UXO_T61_ITEMSUB_IS_FNAME
UXO_TREAT_AS_MIME_SUBJECT
UAL_DEF_MIME_MN_OVERRIDE
UAL_DEF_MIME_MN_OVERRIDE_CS

For more information, see “Configuration Options” on page 301.

Setting Route-Specific Encoding Options for Outgoing Messages

The following configuration options affect the encoding of outgoing MIME messages:

UXO_MIME_FNAME_ENCODING	UXO_MIME_SUBJECT_ENCODING
UXO_MIME_OMIT_DEF_CTENC_HDR	UXO_MIME_SUBJECT_FOLDING
UXO_MIME_SUBJ_NO_SPACE_SEPS	UXO_MIME_TEXTFILE_ENCODING
UXO_MIME_SUBJECT_BENC_NONASCII	UXO_T61_ITEMSUB_IS_FNAME

If you set any of these options in the general.cfg file, they apply to all routes. You can override the settings for a specific route by including the options in a route-specific configuration file.

For any route, Scalix uses the options set in the route-specific file, if the file exists. If an option is not set in the route-specific file, Scalix uses the setting in general.cfg. If the option is not set in general.cfg, then Scalix uses the default value. The route-specific configuration file is:

```
~scalix/sys/route.cfg/routename
```

where routename is the name of the route in printable string format:

```
OU1,OU2,OU3,OU4,O
```

Filenames must be in lower case and cannot include wildcards; you must create a route-specific configuration file for each individual route.

Note that each filename must include the attributes OU1, OU2, OU3, OU4, and O, separated by commas. Null values must be specified.

For example, the route-specific configuration file for the route Moscow,Sales would be:

```
moscow,sales,,,
```

You can see which routes have route-specific configuration files by generating a directory listing of ~scalix/sys/route.cfg.

MIME Encoded Messages with TNEF Attachment

Note TNEF routing is available only in Scalix Enterprise Edition. For more information, see "About Scalix Product Editions".

Incoming MIME-format Internet mail messages that contain TNEF-encoded MAPI properties are recognized by the Internet Mail Gateway and can be converted to a standard Scalix header, one or more body parts, and a WINMAIL.DAT attachment containing the MAPI properties.

The following options in the General Configuration File (~/sys/general.cfg) can be of use in dealing with MIME/TNEF-encoded messages:

- UAL_DEF_TNEF_MN_OVERRIDE

- UAL_DEF_TNEF_MN_OVERRIDE_CS

UUENCODE Encoded Messages

Incoming Internet mail messages conforming to RFC 1154 are recognized by the gateway and automatically converted to a standard Scalix header and one or more body parts. Each body part can be textual, binary, or another message. The following table lists recognized ARPA header encoding keywords.

Keyword	Description
text	Parts with an encoding of "text" become Scalix textual body parts.
message	Parts with an encoding of "message" become nested messages within a Scalix message. Parts within the nested message are decoded according to their keywords.
hex	Parts with an encoding of "hex" are decoded and become Scalix body parts with a file type of "binary" (file code 0). This file type identification can be refined using file type coercion at the Service Router.
uuencode	Parts with an encoding of "uuencode" are decoded and become Scalix body parts with a file type of "binary" (file code 0). This file type identification can be refined using file type coercion at the Service Router.

Matching of keywords is not case sensitive. Unrecognized keywords are treated as "text" and passed into Scalix in their original encoded format.

Incoming Internet mail messages that do not conform to RFC 1154 but contain UUENCODE encoded parts identified by a line beginning "begin access_mode filename" (where access_mode is 3 or 4 octal digits), and ending with end are assumed to be encoded and will be decoded. A decoded part is placed in a body part with a file type of "binary" (file code 0) and the subject filename. The file type identification can be refined using file type coercion at the Service Router. Unlike messages that do conform to RFC 1154, the default mailnode, if required, is that configured for SHAR encoded messages.

To disable the decoding of non-standard RFC 1154 Internet mail messages, use the UXI_UUDECODE_ARPA_TOKEN option.

For outgoing messages, the message format is determined by the route information for that route. If it is uuencode, the gateway generates an Internet mail message that conforms to RFC 1154. The following ARPA header encoding keywords can be generated:

Keyword	Description
text	Used for all text body parts, and body parts converted to text (specified in the `unixout.str` file).
message	Used for all nested messages within a Scalix message. Body parts within the nested message are encoded according to their type (after any conversions specified in the `unixout.str` file have been performed).
uuencode	Used for all non-textual body parts that remain after any conversions and deletions (trashed body parts) specified in the `unixout.str` file have been performed. The parts themselves are encoded using `uuencode`.

SHAR Encoded Messages

Standard, "classic", RFC 822 format Internet mail messages (SHAR encoding) consist of an ARPA header and a single textual body part. Incoming Internet mail messages of this type are automatically converted to a standard Scalix header and textual body part.

For outgoing messages, the message format is determined by the route information for that route. If it is shar, the gateway generates an Internet mail message that conforms to RFC 822 with an extension to support non-textual body parts. Non-textual body parts, that remain after any conversions and deletions (trashed body parts) specified in the unixout.str file have been performed, are encoded and placed at the end of the message. The encoding is by the program shar (shell archive package). Recipients of such messages can decode this portion of the message by copying it into a file and then executing that file.

Body Part Conversion

Steering files are used to specify how message body parts are converted at the Internet Mail Gateway and other gateways. The following steering files are supplied with Scalix:

- mimeout.str
- tnefout.str

You can create the following additional steering files if you require them:

- mimein.str

Scalix automatically uses any of these files that you create.

All Internet Mail Gateway steering files (those supplied and those you create) are held under ~/sys. They are all ASCII files, and can therefore be edited with any text editor.

Note	The formatting of messages read via the POP3 Server and the IMAP4 Server is controlled by the steering file brwmime.str. This file is supplied with Scalix and is identical to mimeout.str.

Steering Files for Outgoing Scalix Messages

The following steering files configure the conversion of message body parts, the character set of textual body parts, and the character set of the ARPA header for Scalix messages routed to the Internet Mail Gateway:

File Name	Description
mimeout.str	Controls the formatting of outgoing messages routed through the Internet Mail Gateway, where the route information is "MIME". This steering file is supplied with Scalix.
sharout.str	Controls the formatting of outgoing messages routed through the Internet Mail Gateway, where the route information is "SHAR". You must create this steering file if you require it.
tnefout.str	Controls the formatting of outgoing messages routed through the Internet Mail Gateway, where the route information is "TNEF". This steering file is supplied with Scalix. It also controls the formatting of outgoing MIME encoded messages with TNEF attachment through the Internet Mail Gateway.

File Name	Description
unixout.str	Controls the formatting of outgoing messages routed through the Internet Mail Gateway, where the route information is "SHAR" or "UUENCODE". This steering file is supplied with Scalix.
uxuuout.str	Controls the formatting of outgoing messages routed through the Internet Mail Gateway, where the route information is "UUENCODE". You must create this steering file if you require it.

Note TNEF routing is available only in Scalix Enterprise Edition. For more information, see "About Scalix Product Editions".

The default for "SHAR" and "UUENCODE" is to convert to text everything that can be, and to convert character sets to IA5. Any non-textual body parts remaining after these conversions are either "SHAR" encoded, "MIME" encoded, or "UUENCODE" encoded depending on the encoding specified for the route.

If the message contains a "Conversion Prohibited" record, no conversions are performed.

Steering Files for Incoming Internet Mail Messages

The following steering files configure the conversion of the character set used in textual body parts and message header strings of incoming Internet mail messages:

File Name	Description
unixin.str	Controls the formatting of incoming messages routed through the Internet Mail Gateway, where the route information is "UUENCODE" or "SHAR". This steering file is supplied with Scalix.
mimein.str	Controls the formatting of incoming messages routed through the Internet Mail Gateway, where the route information is "MIME". You must create this steering file if you require it.
tnefout.str	Controls the formatting of incoming MIME encoded messages with TNEF attachment through the Internet Mail Gateway. This steering file is supplied with Scalix. It also controls the formatting of outgoing MIME encoded messages with TNEF attachment through the Internet Mail Gateway.

The unixin.str steering file is used only if the option UXI_UNIX_MAIL_CHARSET, in the general.cfg file, is set to a valid character set.

If the option UXI_UNIX_MAIL_CHARSET is not set, the gateway uses the first line beginning 1167 in the corresponding *out.str file and reverses the conversion. For example, the default setting in unixout.str is:

```
1167.ISO8859_1

1167.IA5
```

Therefore, if the option UXI_UNIX_MAIL_CHARSET is not set, the incoming character set is assumed to be IA5 and this is converted to ISO8859/1.

If the option UXI_UNIX_MAIL_CHARSET is set, its value is taken as the incoming character set and the conversion specified in unix.in (or mimein.str if it exists for MIME messages) for this character set is performed.

For details about the option UXI_UNIX_MAIL_CHARSET, see “Configuration Options” on page 301.

Message Heading Mappings at the Internet Mail Gateway

The following service elements are retained when messages pass through the Internet Mail Gateway. The mapping of the service elements is based on RFC 1327.

Element	Meaning
Autoforwarded:	Indicates whether the message has been auto-forwarded.
Bcc:	Incoming messages only. Secondary recipient of the message whose name is not disclosed to the other recipients of the message. BCC names are removed from outgoing messages.
Cc:	Secondary recipient of the message.
Date:	Date and time the message was created.
Expiry-Date:	Date and time after which a message is no longer valid or of use.
From:	Designated originator of the message.
Importance:	Importance of the message. can be High, Normal, or Low. If not present, defaults to Normal importance.
In-Reply-To:	Message identification string of the message being replied to.
Latest-Delivery-Time:	Latest delivery time of the message.
Message-ID:	Unique message identification string.
Obsoletes:	Message identification string of a previous message that this message makes obsolete.
Originator-Return-Address:	Postal address of the originator.
Priority:	Priority of the message. can be Urgent, Normal, or Non-Urgent. If not present, defaults to Normal priority.
References:	Message identification string of a previous message that is being cross-referenced.
Reply-By:	Date and time by which the originator of the message would like a reply.
Reply-To:	List of users that the originator recommends that replies are sent to.
Sender:	Originator of the message.
Sensitivity:	Sensitivity of the message. can be Personal, Private, or Company-Confidential. If not present, defaults to not being sensitive.
Subject:	Subject of the message. If an outgoing message, this text is converted to the character set specified in the unixout.str file.

Element	Meaning
To:	Recipient of the message.
X-Scalix-Autoreplied:	Indicates whether the message is an auto-reply message.
X-Scalix-Hops:	The number of times that a message has been handled by a Scalix Service Router.

Internet Acknowledgments

Internet mail users can associate acknowledgments and acknowledgment requests with the messages they send. These are analogous to the Scalix Non-Delivery Notifications (NDNs) and Delivery Acknowledgments.

The Internet Mail Gateway can convert between Internet acknowledgments and their Scalix equivalents. In this way, for example, Scalix can respond correctly to an Internet acknowledgment request, and the resulting acknowledgment can be converted to its Internet form for return to the sender.

Note Internet acknowledgments are only converted to their Scalix equivalents if the acknowledgment request was generated by a UAL client. If the originating client was not a UAL client (for example, an Internet client), the Internet acknowledgment is passed into Scalix as a Message, to be viewed by that client in the appropriate manner.

Mapping Between Internet and Scalix Acknowledgment Requests

Internet mail uses Delivery Status Notification (DSN) requests to request DSN acknowledgments, and sets the Disposition-Notification-To message header to request Message Disposition Notifications (MDNs).

To convert DSN requests to Scalix acknowledgment requests, the Internet Mail Gateway must translate the relevant Extended Simple Mail Transfer Protocol (ESMTP) commands. The mapping between the ESMTP commands and Scalix acknowledgment requests is as follows:

Internet (ESMTP command)	Scalix
RCPT command, NOTIFY=SUCCESS	Delivery Acknowledgment Request
RCPT command, NOTIFY=FAILURE	NDN or NRN Request
RCPT command, NOTIFY=DELAY	Ignored, as there is no Scalix equivalent
RET option set in MAIL command	Return of Contents flag set for NDNs
ENVID parameter supplied in MAIL command	Value is stored in the TFF_TRANSPORT_ID field of the TFR_MSG_INT_ID record
ORCPT parameter specified for recipient	Value is stored in a field in the TFR_RECIPIENT record. In a Delivery Acknowledgment, the value is returned in a field in the TFR_FULL_ACK record.

Mapping Between Internet and Scalix Acknowledgments

Internet mail uses Delivery Status Notifications (DSNs) and Message Disposition Notifications (MDNs) to report acknowledgments. The following table lists the mapping between DSNs and Scalix acknowledgments:

Internet	Scalix
FAILED DSN	Non-Delivery Notification
EXPANDED DSN	AK_DELIVERED_TO_PDL acknowledgment
DELIVERED DSN	AK_DELIVER acknowledgment
RELAYED DSN	No Scalix equivalent. Sent as Message.
DELAYED DSN	No Scalix equivalent. Sent as Message.

The mapping between MDNs and Scalix acknowledgments depends on the direction of the conversion. The following table lists how incoming MDNs are mapped to Scalix acknowledgments.

Internet	Scalix
DISPLAYED MDN	AK_READ acknowledgment No Scalix equivalent. Sent as Message.
DISPATCHED MDN	
PROCESSED MDN	No Scalix equivalent. Sent as Message.
DELETED MDN	AK_DELETE acknowledgment
DELETED:SUPERSEDED MDN	AK_SUB_DLT_OBSOLETED acknowledgment
DELETED:EXPIRED MDN	AK_SUB_DLT_EXPIRED acknowledgment
DELETED:MAILBOX-TERMINATED MDN	AK_SUB_DLT_SUB_TERM acknowledgment
DENIED MDN	No Scalix equivalent. Sent as Message.
FAILED MDN	No Scalix equivalent. Sent as Message.

The following table lists how outgoing Scalix acknowledgments are mapped to MDNs:

Scalix	Internet
AK_AUTOFORWARD acknowledgment	DISPATCHED MDN
AK_READ acknowledgment	DISPLAYED MDN
AK_SUB_DLT_AUTO acknowledgment	DELETED MDN
AK_SUB_DLT_OBSOLETED acknowledgment	DELETED:SUPERSEDED MDN

Scalix	Internet
AK_SUB_DLT_EXPIRED acknowledgment	DELETED:EXPIRED MDN
AK_SUB_DLT_SUB_TERM acknowledgment	DELETED:MAILBOX-TERMINATED MDN

Scalix acknowledgments not listed in the above table are unlikely to be generated, and will be dropped, as there is no equivalent MDN.

Preventing Conversion Between Scalix and Internet Acknowledgments

You can prevent Scalix from converting any MDNs and DSNs to their Scalix equivalents by setting the general configuration option UXI_NO_CONVERT_REPORTS to TRUE. See “Configuration Options” on page 301 for more information on setting general configuration options. This causes Scalix to present all MDNs and DSNs as Messages.

Internet Mail Gateway Commands

The following table lists and describes commands associated with the Internet Mail Gateway.

Command	Description
omconfux	Configure the Internet Mail Gateway
omshowux	Show the configuration of the Internet Mail Gateway

Chapter 5

The Message Store

This chapter describes the message store which is a component of a message handling system that enables individual users to store messages. This chapter includes the following information:

Message Store Overview

The Message Store is where Scalix places all files that constitute the messages of a user. Both those messages delivered to users by the Local Delivery Service and those messages users create are stored. The Message Store is comprised of files in the server file system (not a database).

Each local user has a User Folder which is a separate area of the Message Store that the user can access to receive, read, send, and file their messages.

The Scalix Message Store is structured hierarchically with container items holding messages and item references to their attachments.

A record of which containers are opened and by which process is kept by the Container Access Monitor.

Message Store activity is logged at the container level by the Item Structure Server to enable troubleshooting activities. When a container is attached to or detached from the Message Store, information is logged in both directions of the Message Store hierarchy for the container (if any) in which the current container is stored as well as for any containers or basic items it is holding.

The Single User Restore utility enables the data files of an individual user to be listed, recovered, and restored to the current Message Store.

Message Store Structure

The Scalix Message Store is organized in a hierarchical structure to make access efficient. The structure is composed of:

Container items

Items containing pointers to groups of other items. User Folders, Filing Cabinets, Trays, Folders, and Messages are examples of container items. A container item can hold other container items and basic items.

Basic items

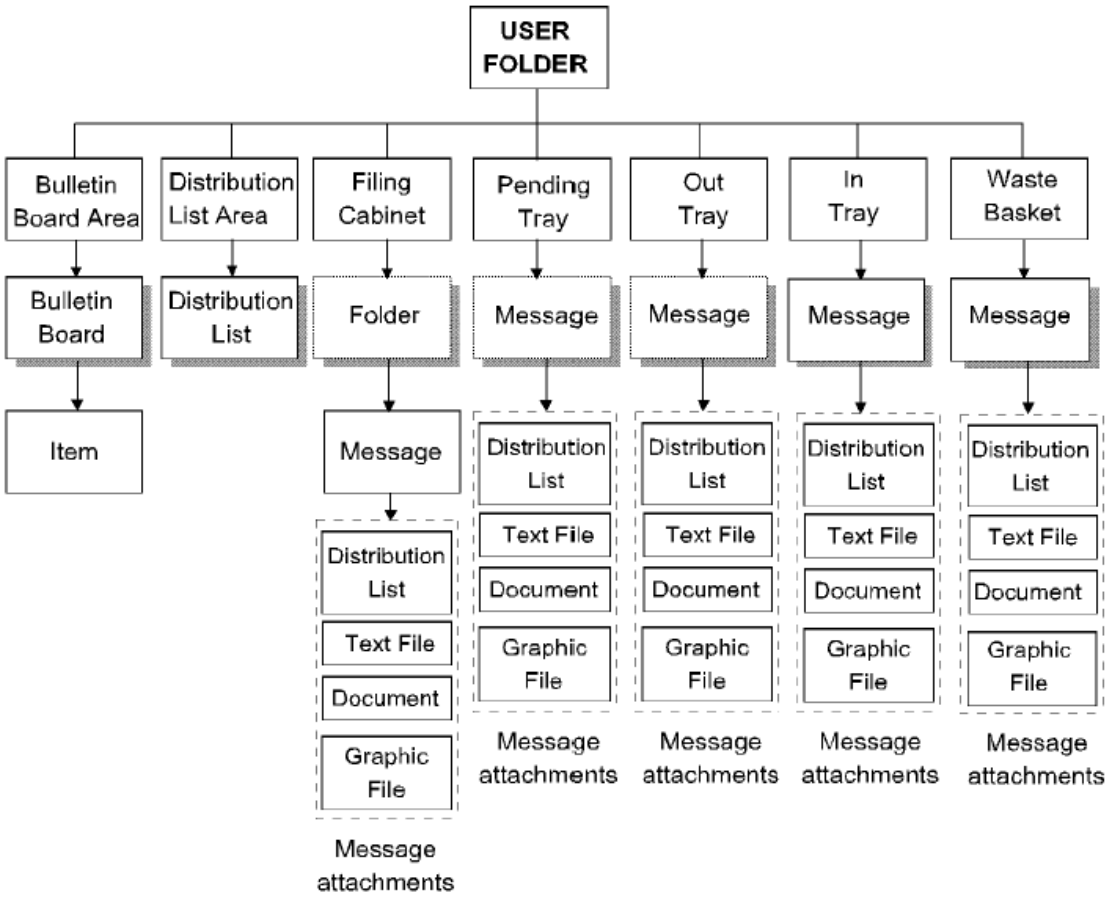
Individual items (typically basic items are attachments to a message). Distribution lists, text, and binary files are examples of basic items. Basic items can be held within container items or can be stored separately within the Message Store.

The following section discusses how these items are used to store user messages in their own area of the Message Store and illustrates the logical structure and physical structure of a User Folder.

User Folders

Each local user has just one User Folder, containing the items associated with that user's messages. The logical structure of a typical Scalix User Folder is illustrated in the following illustration.

User Folder Structure



As shown in the previous illustration, the User Folder is a container item holding several other types of container items. These items are briefly described in the following list.

Trays

- The Filing Cabinet contains item references the user's received and sent messages as well as other items, such as folders, explicitly stored by the user.
- The In Tray contains item references to the user's incoming messages.
- The Out Tray contains item references to the user's outgoing messages.
- The Pending Tray is a tracking area for messages sent by the user. A copy of a sent message is stored in the Pending Tray, while the message itself is sent from the Out Tray. For messages which have acknowledgments set, receipt of the message triggers an acknowledgment receipt, which is then compared with the copy of the message in the Pending Tray.

Folders

- The Bulletin Board Area is a shared folder, which contains item references to any Bulletin Boards configured for the user (specified by the Access Control List) as well as other items, such as folders and messages. See “Public Distribution Lists” on page 139 for more information about Bulletin Boards, “Access Control Lists” on page 197 for more information about ACLs.
- The Distribution List Area is a private folder, which contains item references any user-defined distribution lists.
- The Waste Basket is a private folder, which contains item references to items deleted from User Folder containers by the user. These items are stored in the Waste Basket for a specified period of time before actually being deleted from the Message Store.

Each of these container items, in turn, can reference other types of Message Store items, including:

- Basic items comprising the actual information to be attached to a message, such as: distribution lists, text, word processor documents, graphics images, binary files, spreadsheets, and so on.
- Distribution lists providing delivery information, such as O/R Addresses identifying a user, a group of users, or other distribution lists used to transmit an envelope across the message handling system.
- Folders referencing messages, basic items, and other folders.
- Messages referencing the distribution list and attachments to be transmitted across the message handling system and delivered to recipients.

Users can choose from a selection of clients to access their User Folder. The client interface affects users’ access to their User Folders in the following ways:

- The container items visible to the user, the actual name used for the container items, and the order in which they appear depend on the client interface used. For example, in some clients, the Pending Tray might be called the "Drafts" container, while in other clients it might be an implicit sub-container under the Out Tray.

Some clients do not allow nested folders or do not support Bulletin Boards. However, where these items do exist, their functionality is as described.

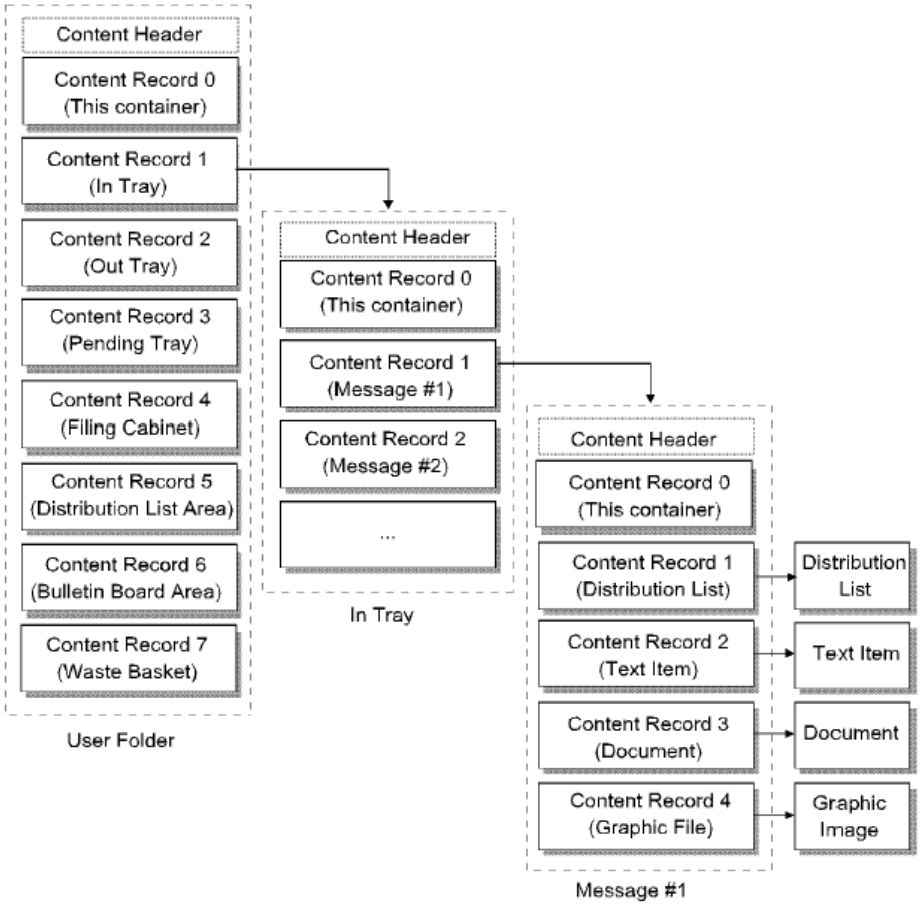
- The client interface might use passwords to control user access to the Message Store. If passwords are set, users must "sign on" with the correct password before they can access their User Folder.

Container Items

Attachments to messages and distribution lists are generally basic items. All other Message Store items are container items, which can store any other types of items. Within the Scalix Message Store, the physical structure of a container item is a serialized file comprising a fixed header and then a variable number of content records, which describe the items attached to the container.

The container header always is content record zero (0) and describes the container itself. Each subsequent content record describes the attachments and includes pointers to the

actual files which hold the information. This physical structure, illustrated in the following figure, enables Scalix to optimize storage.



Container Item Structure

Scalix minimizes the space used to store messages by sharing each message with all local users to whom it is delivered. That is, instead of delivering a copy of the message to each user, it places an item reference to the message in each user's In Tray. Similarly, when a user stores a message from a local user in the Filing Cabinet, it is the item reference, rather than a second copy of the message, that is stored.

Message Store Item Locations

A number of commands are available to manipulate Message Store items and data. These are listed in the section "Message Store Commands" on page 68. Scalix system administrators who need to use some of these commands might need to know the location of User Folder items.

The following container items are stored under ~/user*:

User Folder	Pending Tray
In Tray	Filing Cabinet
Out Tray	Distribution List Area

These other items are stored under ~/data:

- Bulletin Board Area
- Other containers (for example, folders and messages)
- Basic items

Container Access Monitor

The Container Access Monitor is a daemon process (omctmon) that keeps a record of which container items are opened and by which process. It also records size changes that have not yet been written to disk (for example, when an item is deleted).

This daemon process enables Scalix to ensure that the information in containers is valid for the current session.

Rather than writing this information into Message Store files, the Container Access Monitor uses inter-process communication to interact with all Scalix processes. This reduces Message Store input/output (I/O).

Note	To access the Message Store, the Container Access Monitor must run continuously. The process only stops when Scalix is shut down completely.

Item Structure Server

The Item Structure Server is used for repairing the Message Store in the event of corruption or system failure, and for restoring individual user's Message Store data, with the Single User Restore utility.

The Item Structure Server logs all structural changes made to the Message Store, such as attaching and detaching of containers. Information is recorded about a container's parent and child relationships.

Item Structure Server Components

The Item Structure Server consists of the following components:

- Item Structure Server daemon process (omissdm), which records in log files all structural changes to the Message Store.
- Item Structure Server database, which is a dbVista database that stores the Item Structure log files for use by omcontain. By default there is only one database, which is stored in ~/dir/../is/0.

- Item Structure Server log files produced by the Item Structure Server daemon process, and stored in ~/structlog/IS*.
- Item Structure Server log files produced by setting ISL_LOG_IF_OFF=TRUE in the General Configuration file (~/sys/general.cfg).
- Item Structure Server log files produced by the omscan command.
- omupdtis utility, which is used to apply Item Structure Server log files to the database.

The Item Structure Server Daemon

- To start the Item Structure Server daemon, use omon -a iss. The Item Structure Server daemon (omissdm) starts logging all structural changes made to the Message Store. The Item Structure Server log files produced by the Item Structure Server daemon are stored in the ~/structlog directory.
- You can disable the Item Structure Server by using omoff -a iss, or you can disable the flow of data to the Item Structure Server by using the option ISL_DISABLE_LOGGING=TRUE in the General Configuration file (~/sys/general.cfg). However, if you do either of these things, no further structural changes are logged; you will be unable to build a representation of the Message Store's structure for use with the Single User Restore utility.
- Therefore, to use the Single User Restore utility, the Item Structure Server daemon must be enabled. Alternatively, if it is disabled, the option ISL_LOG_IF_OFF=TRUE must be set in the General Configuration file. Setting this option ensures that Structural changes continue to be logged, but directly to the Item Structure Server log files, which results in reduced performance. You must also make sure that the option ISL_DISABLE_LOGGING is not set to TRUE, because this takes precedence over ISL_LOG_IF_OFF=TRUE. For more information about ISL_LOG_IF_OFF and ISL_DISABLE_LOGGING, see "Configuration Options" on page 301.
- To check that the Item Structure Server is enabled, enter omstat -a. The status of the Item Structure Server must be reported as Started.
- Information about structural changes during the time the Item Structure Server was off is permanently lost (unless the ISL_LOG_IF_OFF=TRUE option is set).
- Immediately after restarting the Item Structure Server, you should produce an omscan log file and use it to build an Item Structure Server database containing a representation of the whole Message Store. Use this database as the new initial Item Structure Server database, the start point for subsequent database updates. See the section "Item Structure Server Log Files Produced by omscan" on page 62 for more details.

The Item Structure Server Database

The structural changes logged are applied to the Item Structure Server database to create a representation of the Message Store's structure at a point in time. The information in this database can be used by recovery tools.

This database provides the view of the Message Store to serve as the reference point from which to start updating the database with logged structural changes. You can use omupdtis to apply the structural changes stored in the log files produced either by the Item Structure

Server daemon, or by omscan, whenever it becomes necessary to update the view of the Message Store's structure.

Note	The omscan command cannot run with Scalix shut down. The Container Access Monitor must be running.

You cannot remove structural changes from a database after they have been applied. If you need to restore an Item Structure Server database to its state before an update, you should make a backup copy of the database before proceeding with the update. Then you can restore the backup copy of the database and update it applying different parameters if necessary.

Item Structure Server Daemon Log Files

The Item Structure Server log files produced by the Item Structure Server daemon are stored in the ~/structlog directory. They all have the prefix IS.

The omupdtis command uses these files to create a view of the structure of your Message Store at a specified time. You can specify that they are removed after being applied by omupdtis if you are certain that you will not need to create a view of the Message Store at an earlier time.

Item Structure Server log files are timestamped. They can be applied selectively to the database according to the date and time criteria specified.

You should not update a database with the same Item Structure Server daemon log files more than once.

You should only remove these files when certain that you no longer need them. To remove Item Structure Server log files, use omupdtis. See the online manual entry for details of the omupdtis command.

Consider archiving the Item Structure Server daemon log files separately.

Item Structure Server Log Files Produced by omscan

The omscan -A -a -l file command produces an Item Structure Server log file that can be used by omupdtis instead of the log files in ~/structlog.

When the file produced by omscan -A -a -l is applied to the Item Structure Server database by omupdtis, a database containing a representation of the structure of the whole Message Store is produced.

You can run omscan -A -a -l file while Scalix is running. User activity during the omscan is logged in the Item Structure Server log files in ~/structlog.

Before applying an omscan log file to a database, you should fix any structural errors and rerun the command to create an output file without structural errors.

To update the database with the structural changes in the log file produced by omscan, use the omupdtis command.

After updating the database with the log file produced by omscan, you must ensure that:

- The structural changes logged before the omscan log file was produced are not reapplied.
- The structural changes logged during the omscan are applied.

To do this, update the database with ~/structlog/IS* files timestamped later than the time the omscan command was started. To keep a record of the time the omscan command started, run omscan -t after the command has run and record the output.

The omupdtis Command

The omupdtis command is used to update the Item Structure Server database with the structural changes that have been logged.

By default, omupdtis uses the Item Structure Server log files stored in ~/structlog and applies all the structural changes logged since the last time the omupdtis command ran successfully. To find out at what time the omupdtis command last ran successfully, type omshowis.

There are options available enabling you to specify which Item Structure Server log files to use to update the database. You also can specify that omupdtis use an Item Structure Server log file created by omscan instead of the log files in ~/structlog.

Guidelines for Using the Item Structure Server

After you have created your initial Item Structure Server database, the Item Structure Server continually logs all structural change activity occurring in the Message Store. The Item Structure Server log files are stored in ~/structlog.

You can employ one of the following approaches to updating the Item Structure Server database:

Approach 1

Run omupdtis regularly to maintain the most up-to-date view of the Message Store's structure possible.

This approach is recommended if using the Item Structure Server for troubleshooting corrupt containers and reattaching messages.

Approach 2

Maintain the Item Structure Server database's representation of the Message Store's structure at some time prior to the current time.

This approach is recommended if performing Single User Restore operations. The interval at which you update the Item Structure Server database will be determined by factors such as the following:

- Your backup cycle.
- Number of days structural change history you want to store on disk.
- Amount of disk space available for storing log files.

Using this approach, you would update the Item Structure Server database when required for a Single User Restore operation.

Updating the Item Structure Server Database

You can update the Item Structure Server database in one of the following ways:

- Apply the default Item Structure Server log files.

  To ensure that the representation of your Message Store's structure stored in the Item Structure Server database is accurate, you must ensure that Item Structure Server log files are not applied more than once.
- Initialize the Item Structure Server database and apply the `omscan` Item Structure Server log file.

  If you initialize the Item Structure Server database and apply an `omscan` log file, you must know the exact time that the `omscan` log file was created (not the last modified time).

To find this out, run the omscan -A -a -l filename command, then issue the omscan -t command, then run the omscan command again.

Use this date and time with the omupdtis command. This ensures that Item Structure Server log files generated before the omscan log file are not reapplied as this could create an inaccurate representation of your Message Store.

You can only roll the Item Structure Server database forward. If you need to create views of the Message Store at a time prior to the latest time the Item Structure Server database was updated, you should do the following:

- Keep a backup copy of the Item Structure Server database before each update.
- Keep all Item Structure Server daemon log files for the time period for which you want to build Item Structure Server databases.

When you need to create an earlier view of the Message Store, restore an earlier version of the Item Structure Server database and reapply the Item Structure Server log files, specifying a new, earlier time.

Managing Item Structure Server Log Files

To reduce the number of Item Structure Server log files stored, you can use omscan to create an Item Structure Server log file on a regular basis in line with your backup cycle. You will then only have to keep the Item Structure Server daemon log files generated between each omscan run.

Scalix recommends you use the omupdtis command to remove Item Structure Server log files. Caution is advised if you delete any Item Structure Server log files using an operating system command. If you do use an operating system command to remove Item Structure Server log files, do this only with the Item Structure Server off.

Guidelines for Managing Backups and the Item Structure Server

This section provides general guidelines for managing the Item Structure Server and backups.

Before doing a backup, we recommend you check that the Message Store is in a structurally correct state. To do this, run omscan -a -l filename concurrently with Scalix. If omscan detects errors, you will need to use the appropriate troubleshooting tools, such as omcontain, to fix them. See “Fixing Orphans and Corrupt Containers Reported by omscan” on page 65 for more information.

If omscan reports that the Message Store is structurally correct, back it up using your normal backup tool and include the following:

- The omscan Item Structure Server log file.
- The latest version of the Item Structure Server database.
- The Item Structure Server daemon log files up to the time of the backup

If the omscan command reports that are no corruptions, it is not essential to have the omscan log file. Scalix recommends you include it for safety and completeness. Keeping a copy of the omscan Item Structure Server log file and the Item Structure Server daemon log files up until the time of the omscan means that you will always have the ability to create a complete view of the Message Store’s structure at the time of the omscan.

If you do not keep a backup copy of the omscan log file, you can build the Item Structure Server database for the time of the backup by taking the backed up Item Structure Server database and applying the default Item Structure Server log files.

Fixing Orphans and Corrupt Containers Reported by omscan

If omscan detects orphans, you can need to determine for each one whether or not it should be reattached using omcontain.

This troubleshooting activity is logged by the Item Structure Server, so the information is recorded for future builds of the Item Structure Server database.

If omscan detects a corrupt container, it reports this corruption but does not make an entry for the container in the omscan log file. The omscan log file does not contain information about corrupt containers or anything below them. This means you must use omcontain to recreate the missing containers and reattach the children.

After fixing any errors, you should rebuild the Item Structure Server database. Information about the children of items you reattached is not included. To make sure everything in the Item Structure Server database is correct, run omscan in single-user mode for any user whose Message Store you fixed. For example:

```
omscan -A -U username -l filename
```

The omscan log file produced contains a full representation of the user’s Message Store. You can apply this omscan log file to your Item Structure Server database. It does not matter if it contains some attachments already listed in the Item Structure Server database.

Single User Restore Utility

The Single User Restore utility enables you to prepare a list of files required to restore a single user. Recover the files required and place them in the working directory that you specify. Files can be recovered from the following locations:

- The current Message Store.
- The ~/orphans directory.

To use this utility:

1. Create a data file for the user based on the working directory.
2. Restore the data file to the current Message Store.

The Single User Restore utility relies upon:

- The Item Structure Server which must be running continuously (see "Item Structure Server" on page 60).
- The production of an initial Item Structure Server database.
- The Item Structure Server log files stored in the ~/structlog directory (see "Item Structure Server" on page 60).

Backup Methods Supported by the Single User Restore Utility

You can use Scalix's Single User Restore utility with various backup methods. Backup methods most suitable for use with Single User Restore are those that enable you to recover a very large number (possibly thousands) of specified files.

The following table lists the suitability of some common backup tools:

Tool	Description
fbackup	Allows you to supply a file containing a list of filenames to restore. You will need to create a script to modify the file produced by omprepsur for use with fbackup.
tar	Cannot accept a list of files contained in a single file, but only individual filenames from the command line. Therefore, the command-line length restriction makes this backup method unwieldy if a large number of files is required. In this case, you will need to restore all files from backup and recover the required files from there.
dd	Requires you to restore all files from backup and recover the required files from there.
cpio	Cannot accept a list of files contained in a single file, but only individual filenames from the command line. Therefore, the command-line length restriction makes this backup method unwieldy if a large number of files is required. In this case, you will need to restore all files from backup and recover the required files from there.
OmniBack	Allows you to supply a file containing a list of filenames to restore. You will need to create a script to modify the file produced by `omprepsur` for use with OmniBack.

If the backup method you use does not allow you to easily recover a large number of specified files, you will need to restore all files from backup and create a script to recover the required files from there.

The omgetsur script provided with the Single User Restore utility is comprehensively commented, enabling you to use it as a template for creating a script for the backup tool you use.

The UserInfo.log File

The Single User Restore utility uses information stored in the UserInfo.log file.

Whenever the administrative commands omaddu, ommodu, and omscan are executed, the information in the User Information Log file (~/sys/UserInfo.log) is updated. The UserInfo.log file contains the following details about each user:

- Scalix O/R Address
- Scalix ID
- Linux login ID
- Capabilities
- Aliases
- Language

The UserInfo.log file is a binary file. You should never remove or modify this file.

Message Store Commands

The following table lists and describes commands associated with the Message Store.

Command	Description
omcontain	Manipulate containers in the Message Store
omcpinu	Copy a user's Message Store data from a file
omcpoutu	Copy a user's Message Store data to a file
omdosur	Create a data file for restoring a single user
omdref	Convert a Scalix DirectRef into a readable description of the item represented, including Message Store item hierarchy
omdumpis	Write Item Structure database to standard output
omgetsur	Get files from an archive
omlimit	Set Message Store size limits globally or for a user
omnewis	Create an empty database
omprepsur	List files required for single user restore
omscan	Scan, report, and repair Scalix data inconsistencies
omshowis	Display the date `omupdtis` was last run
omsnoop	Report on potential Message Store conversion problems
omsuspend	Halt all client activity temporarily
omtidyallu	Delete items from the Message Store
omtidyu	Delete items from the Message Store for an individual and search nested folders for an item to delete
omupdtis	Read Item Structure log entries and update the database
tfbrowse	Convert between Scalix transaction file format and textual format

Chapter 6

The Service Router

This chapter describes the Service Router, and includes the following information:

About the Service Router

All messages arriving on the system or generated by the system pass through the Service Router. The Service Router determines which delivery services are required to deliver a message on to its next "hop". The next hop can be any Scalix interface or gateway, the Local Delivery Service, or the RESOLVE queue. If a message is addressed to several recipients, several services might be required to deliver the message on to its next hop.

Messages are passed to the Service Router through its input queue named ROUTER. The service name is router and the process name is service.router.

How the Service Router handles messages

When the Service Router receives a message, it performs the following tasks:

- Checks the file types in the message body parts and performs any specified file type coercions (see "File Type Coercion" on page 77).
- Updates the active recipient list of the message (see the section "Route Filtering" on page 78).

  If the O/R Address in the message is routed to the RESOLVE queue, the Service Router resolves the address before performing any further routing (see the section "Resolve Queue" on page 79).
- Checks any ruleset specified for the route to match specified attributes against the contents of the message.

If all of the attributes specified in a rule within the ruleset are matched, it performs the action defined in the rule (see "Message Delivery Rulesets" on page 80).

If the defined action is to reject the message, the Service Router performs no further routing for the message.

If the defined action is to defer delivery of the message, the Service Router places the message on the DMM queue (see "Deferred Mail Manager" on page 89). The Deferred Mail Manager handles any subsequent routing of deferred messages.

If other actions are defined, the Service Router performs the appropriate routing.

- Adds a route for each recipient in the active recipient list (see the section "Address Resolution" on page 74).

  If the active distribution list contains BCC recipients, the message is split into separate messages, each with a different active distribution list (one for each BCC recipient, and one for all other recipients).

- Checks the Routing Table to match each recipient address given in the active recipient list with the address pattern and associated delivery service and then routes the message (see "Routing Table" on page 71).

  If the message cannot be routed, the Service Router discards the message and generates a Non-Delivery Report for each recipient address that could not be routed. It returns the Non-Delivery Reports to the originator of the message.

- Checks that the message is not "looping" (see "Loop Detection" on page 90).

  If the message is looping, the Service Router discards the message and generates a Non-Delivery Report for each address in the active recipient list. It returns the Non-Delivery Reports to the originator of the message.

- Checks that the originator of the message has permission to use each delivery service listed in the active recipient list (see "Access Control Lists" on page 197).

  If the originator does not have permission to use a delivery service, a Non-Delivery Report is generated for each recipient address that cannot be delivered to. The Non-Delivery Reports are then returned to the originator.

- Attaches the message to each delivery service queue listed in the active recipient list.

Note	If a message is routed to a gateway and the recipient address does not contain a foreign address in the DDA, the Service Router checks the system default Directory to see if a foreign address can be supplied for that O/R Address.

Routing Table

The Routing Table is used by the Service Router to determine which delivery service queues a message is attached to. The table is in the ~/sys/router file. An example routing table is shown below:

Queue	Address	Information
UNIX	ny,hq,internet	MIME
LOCAL	ny,hq,mis	
LOCAL	ny,sales,mis	
SMINTFC	boston,factory,*,*	scalix@jupiter.com
SMINTFC	ny,factory,*,*	scalix@saturn
OMX400	*,*,*,*/pinewood/fr/atlas/forester	DEFAULT
OMX400	*,*,*,*/pinewood/gb/gold_400/forester	DEFAULT

The Routing Table contains:

- a list of address patterns
- the delivery service queue name associated with each address
- (optionally) information specific to that delivery service

For example, a message with a recipient address of Ed Gonzalez jnr./ny,factory,mis matches the Routing Table address pattern ny,factory,*,*. Using the information in the Routing Table for this address pattern, the Service Router attaches the message to the SMINTFC queue (the queue for the Sendmail Interface), and specify scalix@saturn as the internet mail address of the recipient Scalix system.

Entries are automatically added to the Routing Table when local mailnodes are added or when routes to local gateways and interfaces are added.

Use the commands omaddrt, omdelrt, ommodrt, and omshowrt to add, delete, modify, and list routes configured in the Routing Table, respectively.

Routing Table Address Pattern Rules

The O/R Address attributes on which a message is routed are hierarchically ordered:

An attribute must be fully specified, partly represented with a wildcard, wholly represented with a wildcard, or left blank.

The following table lists routing table address pattern rules.

	Order Address Forms				
Order	Mnemonic	Numeric	Postal (formatted)	Postal (unformatted)	Terminal
1	C	C	C	C	C
2	A	A	A	A	A
3	P	P	P	P	P
4	O		PD-SN	PS-SN	X121
5	OU1		PD-C	PD-C	
6	OU2		PD-PC	PD-C	

Order	Order Address Forms				
	Mnemonic	Numeric	Postal (formatted)	Postal (unformatted)	Terminal
7	OU3				
8	OU4				

Wildcards in the Routing Table

You can widen the scope of address patterns in the Routing Table by using wildcards. The wildcard character is an asterisk (*). The wildcard represents zero or more characters.

Using wildcards enables you to have a smaller table that is easier to maintain. Addresses can be added or deleted on other systems without you having to change your Routing Table.

To partly represent an attribute with a wildcard, place the wildcard character at the end of the partial value. For example:

- O=Pinewood is fully specified
- O=Pine* is partly represented with a wildcard
- O=* is wholly represented with a wildcar*d*

If an attribute contains a wildcard, all less significant attributes must be wholly represented with wildcards or left blank.

If an Organizational Unit Name is left blank, all less significant Organizational Unit Names must be also left blank.

For example, the following entry in a Routing Table routes all messages with recipient addresses that match the address pattern through the Sendmail Interface to another Scalix system on a machine named saturn:

SMINTFC	ny,factory,*,*	scalix@saturn

The following recipient addresses will match this address pattern (and any addresses of a similar form that are subsequently added on other systems):

```
Ed Gonzalez Jnr/OU1=ny/OU2=factory/OU3=mis
Linda McLeod/OU1=ny/OU2=factory/OU3=finance
Lance Noguchi/OU1=ny/OU2=factory/OU3=quality
```

Without the use of wildcards in the address pattern, three entries would be required in the Routing Table for the recipient addresses:

```
SMINTFC     ny,factory,mis        scalix@saturn
SMINTFC     ny,factory,finance    scalix@saturn
SMINTFC     ny,factory,quality    scalix@saturn
```

Note	The Organization Name and Organizational Unit Name attributes can be specified in either printable strings or teletex strings, or both. If both forms are specified and one form of the attribute is represented by a wildcard, then the other form must also be represented by a wildcard to the same extent.

Examples of Routing Table Address Patterns

The address pattern

```
OU1=ny/OU2=hq/OU3=linux
```

is valid, with the first three Organizational Unit Names being fully specified.

The address pattern

```
OU1=ny/OU2=*/OU3=*/OU4=*
```

is valid and represents *all* addresses containing only a Personal Name and Organizational Unit Names where OU1 equals ny. However, the address pattern

```
OU1=*/OU2=*/OU3=quality/OU4=*
```

is invalid because OU1 and OU2 are wholly represented by wildcards, so OU3 also must be wholly represented by a wildcard.

The address pattern

```
OU1=sal*/OU2=*/OU3=*/OU4=*
```

is valid and represents all addresses where OU1 begins with the characters `sal`. The following addresses match this pattern:

```
OU1=sales/OU2=southern/OU3=export
OU1=sales/OU2=accounts OU1=salem/OU2=legal
```

The address pattern

```
OU1=*sal/OU2=*/OU3=*/OU4=*
```

is invalid because the wildcard character is incorrectly positioned in OU1. It must be placed at the end of the partial value.

The address pattern

```
OU1=*/OU2=*/OU3=*/OU4=*/O=*/P=*/A=*/C=fr
```

is valid and represents all addresses of this form that specify a Country Name of `fr`. However, the address pattern

```
O=pinewood/P=*/A=*/C=fr
```

is invalid because the ADMD and PRMD are wholly represented by wildcards, so the Organization Name, being of lesser significance, must also be represented by wildcards.

The address pattern

```
OU1=*/OU2=*/OU3=*/OU4=*/P=forester/A=atlas/C=fr
```

is valid and represents all addresses of this form that specify a Country Name of `fr`, an ADMD of `atlas`, a PRMD of `forester`, and no Organization Name. However, the address pattern

```
OU1=*/OU3=*/P=forester/A=atlas/C=fr
```

is invalid because OU2 has been left blank, so OU3 must also be left blank.

The address pattern

```
O=*/P=forester/A=atlas/ì
```

is valid and represents all addresses of this form that specify a Country Name of `fr`, an ADMD of `atlas`, a PRMD of `forester`, and any Organization Name (teletex and printable string, because an attribute wholly represented with a wildcard matches to both attribute types). Additionally, the address pattern

```
O=pine*/O-TX=*/P=forester/A=atlas/C=fr
```

is invalid because, where an attribute is represented in a printable string form and a teletex string form, both attributes must be specified to the same extent.

Routing Table Search Rules

When the Routing Table is searched for an address pattern that matches the address of the recipient, these rules are used:

- Match characters regardless of whether they are uppercase or lowercase.
- Ignore address attributes that are not used for routing. (See “Routing Table” on page 71 for a list of attributes that can be routed.)
- Match each attribute:

  A specified attribute matches if each character compares "one for one".

  An attribute partly represented with a wildcard matches if each character in the specified part of the attribute compares one for one.

  A blank attribute matches if the attribute is also blank in the address of the recipient.

  An attribute wholly represented by a wildcard matches anything.
- If only one address pattern matches the address of the recipient, use the routing information for the address pattern.
- If a number of address patterns match the address of the recipient, select the "best fit" address pattern. Evaluate the quality of the match for each attribute as follows:

  First: a match to a specified attribute.

  Second: a match to an attribute partly represented by a wildcard.

  Third: a match to a blank attribute.

  Fourth: a match to an attribute wholly represented by a wildcard.

Start at the most significant attribute; continue evaluating the attributes until the best fit address pattern is found. Use the routing information for this address pattern.

Address Resolution

When the Service Router picks up a message from its queue, it scans the active recipient list to determine where to route the message. The Service Router tries to match the addresses against the address patterns listed in the Routing Table.

If any messages cannot be routed (for example, the active distribution list contains an incomplete O/R Name), the Service Router searches the system default Directory and tries to expand the O/R Name by replacing it with the full O/R Name, including the mailing

address (O/R Address), obtained from a Directory entry. This expansion of an address in an active distribution list is referred to as “O/R Address resolution”.

Once an address is resolved, the message is then sent on to the next destination in the delivery route.

O/R Address resolution is performed by:

- the Service Router for all messages passing through Scalix
- the Local Delivery Service for messages from local Scalix users
- the Client Interface for messages received from external systems

O/R Address Resolution by the Service Router

The Service Router determines which delivery services are required to deliver a message on to its next destination on the way to its specified recipients. When a message passes through the Service Router, it checks if an address in the active recipient list of the message can be routed to a delivery service.

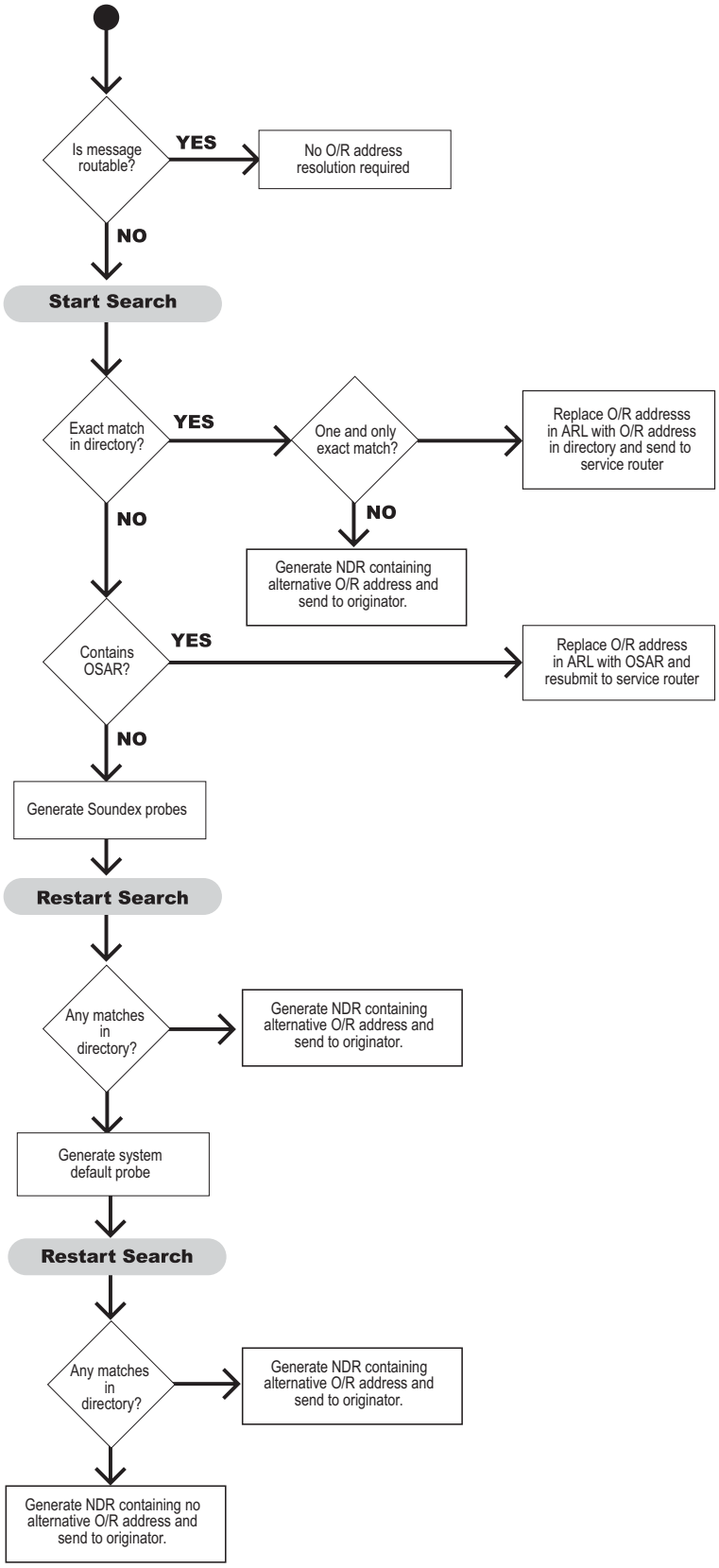
The Service Router might not be able to route a message to an address for several reasons. For example, only the Personal Name is provided, or an O/R Name contains only a Directory Distinguished Name (DDN) (see “Address Resolution” on page 74 for information about DDNs).

If the Service Router can route the address, no address resolution is required and the Service Router sends the message on to the appropriate queue.

If the Service Router cannot route the address, the Service Router tries to resolve the address by searching the system default Directory.

Address resolution performed by the Service Router is displayed in the following illustration.

O/R Address Resolution by Service Router



The following table lists and describes each phase of the Service Router search process.

Search Item	Search Result	What the Service Router Does
Attributes of O/R Address in active recipient list (ARL)	Exact match (except for capitalization) with one entry	Replaces the O/R Address in the active recipient list with the O/R Address found in the Directory and resubmits the message to the Service Router.
	No matches	Replaces O/R Address in active recipient list with the Originator Specified Alternative Recipient (OSAR) (if one is specified in the message) and resubmits to the Service Router.
	More than one exact match	Generates a Non-Delivery Report containing alternative O/R Addresses.
	No exact matches	Generates soundex probes for the given name (with a trailing wildcard) and the surname of the recipient address. (The surname soundex probe created also uses a trailing wildcard.)
Attributes in soundex probes	Any matches	Generates Non-Delivery Notification containing alternative O/R Address and sends to message originator.
	No matches	Generates a system default probe (which itself defaults to the surname soundex probe).
Attributes in system default probe	Any matches	Generates Non-Delivery Report containing alternative O/R Address and sends to message originator.
	No matches	Generates Non-Delivery Report containing no alternative O/R Names and sends to message originator.

File Type Coercion

The file type associated with a body part can be changed by the Service Router.

For example, if a user or gateway identifies a body part as being of binary format when it is really a specific word processor file, the file code associated with that body part can be changed to reflect its true file type. The ~/sys/map.types file specifies the file type coercions to be performed. The ~/nls/language/filetype file lists available file types and their associated file codes.

When the Service Router is started, it attempts to open the map.types file. If the file cannot be opened, no coercions are performed. If it fails to parse part of the file, an error is logged in the Event Log and no coercions are performed.

When the Service Router processes a message, it examines the file types associated with the message body parts (except the active distribution list). If the content of a message body part and its file code matches a line in the map.types file, the file code of the body part is changed.

The format of the `map.types` file is as follows:

```
old-file-code new-file-code[!]pattern
```

Note the following information about file type coercion

- Comment lines (lines starting with a #) are ignored.
- If the file code of the body part matches the old-file-code, then the remainder of the line is examined.
- If pattern matches the content of the body part, then the body part is marked as being of file code new-file-code.
- If pattern is preceded by the exclamation point (!), and pattern does not match the content of the body part, then the body part is marked as being of file code new-file-code.
- pattern has the following format:

  {[@[SUBJECT|*offset*]] {[0x*nn*|"*string*"]}*}*

  The at character (@) followed by the string SUBJECT and a quoted string means "match the quoted string with the subject of the body part".

  The at character (@) followed by a byte offset in decimal means the pattern that follows begins at the indicated byte offset in the body part.

  The string 0xnn specifies a single byte in hexadecimal form. Several of these can occur on the line, interspersed with character strings enclosed in double quotation marks (").

Route Filtering

Route filtering is the process by which the active recipient list is updated as a message passes from one Scalix system to another. For example, a message is routed by the Service Router. Its active recipient list is as follows:

```
Ed Gonzales jnr./ny,factory,mis                SMINTF scalix@saturn
Mark R. Holzmann /boston,factory,mis           SMINTFC scalix@jupiter.com
Chris Wolf/ny,hq,mis                           LOCAL
```

When the Sendmail Interface processes the message, it processes only recipient addresses that are marked with its queue name (SMINTFC). Recipient addresses not routed through the Sendmail Interface are ignored.

When a copy of the message reaches Scalix on jupiter, all the recipients except Mark R. Holzmann have been dealt with by the previous Scalix system and should no longer be in the active recipient list.

To update the active recipient list, a "filter route" is added to the message by the xport.in process on jupiter. The filter route is the queue name of the delivery service and any information associated with it in the active recipient list.

In the above example, the filter route added to the message when it arrives on jupiter is SMINTFC:scalix@jupiter.com.

The Service Router updates the active recipient list by matching the filter route with the recipient addresses in the current active recipient list. Recipient addresses which do not match the filter route are removed from the active recipient list.

Resolve Queue

Route O/R Addresses to the RESOLVE queue to force the Service Router to resolve the address before any further routing is performed.

Because you cannot use wildcards in address patterns routed to the Local Delivery Service queue LOCAL, use of the RESOLVE queue simplifies the configuration of hierarchic addressing in a hub topology. For example, in the following Routing Table, a message with an active recipient address of Mark R. Holzmann/boston is routed to the hub system on mars; only then is it resolved before being routed back to the local system for delivery.

The following displays a Routing Table with no Route to the RESOLVE Queue

```
SMINTFC                 *,*,*,*                 scalix@mars.com
LOCAL                   boston,factory,mis
LOCAL                   boston,factory,admin
```

For details of wildcards in address patterns, see the section “Routing Table Address Pattern Rules” on page 71. For details of the Routing Table, see the section “Routing Table” on page 71.

If the O/R Address pattern boston,*,* is routed to the RESOLVE queue on the local system, as shown in the following Routing Table, then the recipient address Mark R. Holzmann/boston is resolved on the local system by the Service Router before being routed further. Assuming that the Service Router finds only one entry in the local system default Directory that matches the address Mark R. Holzmann/boston, the recipient address is replaced by the one found in the Mark R. Holzmann/boston,factory,mis Directory. The Service Router can then route the message to the Local Delivery Service queue LOCAL.

The following displays a Routing Table with a route to the RESOLVE queue

```
SMINTFC                 *,*,*,*                 scalix@mars.com
RESOLVE                 BOSTON,*,*
LOCAL                   boston,factory,mis
LOCAL                   boston,factory,admin
```

Message Delivery Rulesets

The Service Router determines if any messages require special handling (for example, to be rejected or have their delivery deferred) by checking the settings of attributes in any ruleset specified for a given route. Each rule within a ruleset defines the conditions under which the Service Router should reject or defer the delivery of a message. Rules also specify how the Service Router should handle these messages and the time at which it should perform the specified action.

The contents of the message are compared against each rule within the ruleset to check for a match:

- If all the attributes for a rule are matched, the Service Router performs the action defined for the rule.

  If the defined action is to defer the message, it submits the message to the Deferred Mail Manager queue DMM (see the section “Deferred Mail Manager”).

- If not all of the attributes in the ruleset are matched, by default the Service Router continues with its procedure to route the message (see the section “How the Service Router handles messages” on page 69).

To configure the Service Router to verify a ruleset for a particular route, associate the ruleset with the route using either the omaddrt command (for new routes) or ommodrt (for existing routes). For example, to associate the ruleset off-peak with an existing route of remote,sales, enter the following command:

```
ommodrt -m "remote,sales" -d off-peak
```

There are two reserved ruleset names:

- **ALL-ROUTES:** If this ruleset exists, it applies to all routes, except routes for which you configure a specific ruleset.
- **ALL-ROUTES.VIR:** This ruleset enables virus protection for the Scalix system. If this ruleset exists, Scalix executes the ALL-ROUTES.VIR ruleset before all other rulesets. See “Configuring Scalix Virus Protection” on page 207 for more information.

Note	Rulesets apply to a message only if you associate the recipient of a message with a route for which you configure a ruleset.

Although some rules within a ruleset are associated to the sender of a message (for example the OMLIMIT-EXCEEDED rule), the rule s apply only to messages sent by (for example) user1 to recipients associated with routes which have the ruleset configured, and not all messages sent by user1.

You can see which ruleset (if any) is associated with a given route by using the omshowrt command. For details, see the section “Routing Table Commands” on page 91.

Note	When you specify a ruleset to a Scalix command, Scalix checks the contents of the ruleset and reports any syntax errors. If you change any rules in a ruleset, you are recommended to run the ommodrt command to verify the syntax of the new rule is correct.

Ruleset File Format

A ruleset is a text file you create. There are no restrictions on the filename you give your ruleset; however, rulesets must be stored under the following directory:

A ruleset is a text file you create, and you can enter any name for the ruleset. However, the names "ALL-ROUTES" and "ALL-ROUTES.VIR" have special meanings. You must store rulesets in the following directory:

```
~/rules
```

For example, a ruleset with a filename of off-peak would be stored as:

```
~/rules/off-peak
```

A ruleset consists of a series of text lines. Lines that are blank or start with a hash character (#) are regarded as comment lines. If an argument contains white space, enclose the argument in double quotation marks ("). If an argument contains a double quotation mark (") or a backslash character (/), precede the character with a backslash character.

Each rule in a ruleset must be defined on a single line. A rule contains a number of attributes specified as TAG=value pairs, which define the criteria to be matched in a message.

The following table lists the three categories of ruleset attributes:

Category	Description
message_filter	Is one or more of the attributes defining the parts of the message against which the Service Router checks for matches when processing the message. If no message filter attributes are defined for a rule, then all attributes are considered a match.
day_time	Is one or both of the attributes defining the period during which the Deferred Mail Manager should perform the specified action on the deferred message. These attributes apply only when the DEFER action is defined in the rule.
action_info	Is one or more of the attributes defining the action to be performed or the information to be supplied when the values of the *message_filter* and *day_time* attributes are matched.

See "Examples of Message Delivery Rules" on page 86. for examples of rule sets.

Note	Any files specified as input to ruleset attributes must be stored in the ~/rules directory.

Message Filter Attributes

These attributes define the parts of a message that, when matched, cause the specified action to be performed. These attributes are optional and if none are specified, the Service Router assumes a match for all attributes.

The following table lists the message_filter attributes:

Attribute	Description
BCC-COUNT	The number of BCC addressees that can be contained in a message. This is matched when the number of BCC addresses in the message is greater than or equal to the value specified for this tag. For example, BCC-COUNT=10 causes all messages that have 10 or more BCC addresses to be deferred or rejected, depending on the defined action.
DL-COUNT	The number of addressees that can be contained in the primary Distribution List of a message. This is matched when the number of addresses in the Distribution List is greater than or equal to the value specified for this tag. For example, `DL-COUNT=100` causes all messages that have 100 or more addresses in the primary Distribution List to be deferred or rejected, depending on the defined action. Note that a Distribution List can contain one or more Public Distribution Lists. Each such PDL is initially counted as just one address by the Service Router. However, when it is expanded by the Local Delivery service and passed back to the Service Router, each individual address in the PDL will add to the number of addresses in the primary Distribution List, and so can cause the message to be deferred or rejected.
OMLIMIT-EXCEEDED	A percentage indicating how full a message store component is in relation to its configured limit. A number of message store size limits can be configured for users: overall message store, In Tray, Pending Tray, and so on. See the man page for omlimit for more information. Any NOTIFY action associated with this attribute is executed if the sender of a message has not already been notified within the last day. You can change the default value of 1 day by configuring the OMLIMIT_MIN_WARN_INTERVAL parameter in the general.cfg file. This filter is matched when the sender has a message store component which is at the specified percentage of its configured limit. For example, a value of 100 would match all messages from senders who had exceeded a limit by any amount; a value of 110 would match all messages from users who had exceeded a limit by 10 percent or more. A value less than 100 can be used to match messages from senders who are near to, but not yet at, one of their limits. For example, a value of 90 would match messages from senders who were at 90 percent or more of a limit. See “About the Service Router” on page 69 for more information about configuring message store limits and creating rules to implement sanctions.
ORIGINATOR	pattern A Scalix Address pattern to match against the originator of the message. !filename[:charset] A separate file containing one or more Scalix Address patterns to match against the originator of the message. The optional charset attribute specifies that the text of the file uses a character set other than IA5. The appropriate character set conversion is performed. Address patterns must observe the format and rules for Access Control List address patterns (see “Matching Addresses to O/R Address Patterns in ACLs” on page 201).
PRIORITY	HIGH An urgent message MEDIUM A normal message LOW A non-urgent message

Attribute	Description
RECIPIENT-SERVICE-LEVEL	The service level of the recipient of a message. Service levels are assigned to users solely to enable receipt and delivery rules to be constructed. A value of `0` would check for those recipients for which a service level has not been created.
SENDER-SERVICE-LEVEL	The service level of the sender of a message. Service levels are assigned to users solely to enable receipt and delivery rules to be constructed. A value of 0 would check for those senders for which a service level has not been sent.
SIZE	The size in KBs of a message (this is matched when the message size is greater than or equal to the value specified for this tag).
SUBJECT	pattern The string of text to match against the subject of the message. Wildcard characters (*) can be used. The entire subject line of the message is compared with the *pattern* for a match. This comparison is case sensitive. !filename[:charset] A separate file containing the string of text to match against the subject of the message. The optional charset attribute specifies that the text of the file uses a character set other than IA5. The appropriate character set conversion is performed. The entire subject line of the message is compared with the specified string of text for a match. This comparison is case sensitive. @script[:charset] A separate script or program containing predefined protocols syntax to communicate with the Service Router and the Deferred Mail Manager to match the contents of the subject of a message. *script* must be a readable and executable file. The optional *charset* attribute specifies that the script (or program) uses a character set other than IA5 for the display of the subject. The appropriate character set conversion is performed, and the script (or program) checks for the contents of the subject in the display character set. The script returns a status of matched, or not matched, based on the predefined protocols syntax. A sample script is provided in the section “Example Script for SUBJECT Attribute” on page 88.
VIRUS-FOUND	Specifies whether a message contains a virus. A value of `0` indicates that a virus was not found in the message; a value of `1` indicates that a virus was found. Include this attribute in a rule to enable virus scanning, but not virus cleaning. See “Virus and Spam Protection” on page 205 for more information on virus protection. This attribute can only be used in the ruleset `ALL-ROUTES.VIR`, and applies to all routes. It cannot be applied selectively to specified routes.
VIRUS-UNCLEANED	Specifies whether a message that contains a virus could not be cleaned. A value of `0` indicates that the message was checked and was successfully cleaned of any viruses that were found. A value of `1` indicates that the message contained a virus that could not be cleaned. Include this attribute in a rule to enable virus cleaning. See “Virus and Spam Protection” on page 205 for more information on virus protection. This attribute can only be used in the ruleset ALL-ROUTES.VIR, and applies to all routes. It cannot be applied selectively to specified routes.

Day/Time Ruleset Attributes

These attributes define the period during which the Deferred Mail Manager should defer messages when all other rule attributes are matched and ACTION=DEFER is specified. See the section “Deferred Mail Manager” on page 89 for more information.

The following table lists the day_time attributes.

Attribute	Description
DAY	The day of the week on which to defer messages. This attribute is specified with an integer where 0=Sunday, 1=Monday, 2=Tuesday, etc. You can specify a range of days (for example, 1-5 for Monday through Friday) If no value is specified, all week is assumed.
TIME	The time of day the DEFER action should be performed. This attribute is the local time specified in HH:MM or HH format, using the 24-hour clock (00:00 or 00 is midnight). This attribute must be specified. You can specify a duration of time during which deferred messages can be delivered using the following syntax: time_interval controls the start and stop times during which actions should be performed specified in HH:MM-HH:MM format or in HH-HH format. You also can specify a time at which to begin storing a batch of deferred messages and an interval of time during which to deliver the deferred messages using the following syntax: batch_spec controls the time to start batching messages and the interval during which to deliver the batch of deferred messages specified in HH:MM@HH:MM format or in HH@HH format.

Action/Information Ruleset Attributes

These attributes define the action to be taken or the information to be provided when the message_filter and date_time attributes are matched.

The following table lists the action_info attributes.

Attribute	Description
ACTION	An ACTION attribute must be specified for a rule. For actions which do not automatically return a Non-Delivery Notification, you can specify a message to be returned to the user with the NOTIFY tag. • ALLOW Route the message immediately. • DEFER Defer delivery of the message during the period specified by the *day* and *time* attributes. • DISCARD Discard the message without returning a Non-Delivery Notification to the originator. • REJECT Do not route the message and return a Non-Delivery Notification to the originator. • RETURN Do not route the message and return a Non-Delivery Notification and the original message to the originator.
NAME	The name of a rule within the ruleset. This name is used in any notification message generated and written to the Scalix Event Log and is useful in determining which rule is being applied when a message is routed. This tag is optional.
NDN-INFO	Enables you to replace the standard text string in the supplementary information field with the specified text when Non-Delivery Notification is required. This tag is optional. NDN-INFO is one of: • *text* The text to be included in a Non-Delivery Notification. • *!filename[:charset]* A separate file containing the text to be included in a Non-Delivery Notification. The optional *charset* attribute specifies that the text of the file uses a character set other than IA5. The text is converted to IA5.

Attribute	Description
NOTIFY	Enables you to include supplementary text in the standard notification message returned to the originator when the defined action has been performed on the message. This tag is optional. When NOTIFY is specified, the supplementary text is imported as a text part and attached in front of the standard notification message text. NOTIFY is one of: • No value The standard notification message text containing details of the ruleset applied and the recipients affected is used if NOTIFY is specified with no value. • *text* The supplementary text to be included in a message to the originator. The maximum size of *text* is 255 bytes. • *!filename[:charset]* A separate file containing the supplementary text to be included in a message to the originator. The optional charset attribute specifies that the text of the file uses a character set other than IA5. The appropriate character set conversion is performed. There is no restriction on the size of text contained in filename. NOTIFY is typically used with the ACTION=DEFER or ACTION=DISCARD attributes, since Non-Delivery Notification is not returned to the originator of the message.

Examples of Message Delivery Rules

This section contains example rules illustrating how rules are specified within a ruleset and describing the action taken when the rules are matched by a message.

Deferring Messages Based on Originator

```
PRIORITY=LOW ORIGINATOR="*/CN=*/lab" ACTION=DEFER TIME=11:00-18:00
```

Delivery of all low priority messages sent from the lab mailnode is deferred between 11 a.m. and 6 p.m. (that is, these messages are delivered after 6 p.m. and before 11 a.m.).

Delivering Messages Based on Subject

```
PRIORITY=MEDIUM SIZE=1000 SUBJECT="PUBLIC:*"
ACTION=REJECT ACTION=ALLOW
```

Messages with a subject starting with the text "PUBLIC:", of normal priority and of a size 1 MB or greater are not routed, and a Non-Delivery Notification is returned to the message originator. Any other messages are routed without delay according to the normal Service Router routing process.

Delivering Messages Based on Priority

```
PRIORITY=HIGH     ACTION=ALLOW
PRIORITY=MEDIUM   ACTION=DEFER      TIME=09:00-17:00  DAY=1-5
PRIORITY=LOW      ACTION=DEFER      TIME=06:00-21:00  DAY=1-5
```

Any high priority messages are delivered Sunday through Saturday without delay according to the normal Service Router routing process. Any medium priority messages are submitted to the Deferred Mail Manager for deferral between 9 a.m. and 5 p.m., Monday through Friday. Any low priority messages are submitted to the Deferred Mail Manager for deferral between 6 a.m. and 9 p.m., Monday through Friday.

Rejecting or Deferring Message Delivery Based On Size

```
NAME=Reject_1Mb     SIZE=1000      ACTION=REJECT
NAME=Defer_100Kb    SIZE=100       ACTION=DEFER        TIME=22:00-06:00
```

Any messages of a size 1Mbyte or greater are rejected, and a Non-Delivery Report is sent to the message originator. The delivery of any messages 100 KBs in size (but less than 1Mbyte) is deferred between 10 p.m. and 6 a.m.

Providing Supplementary Text for a Non-Delivery Notification

```
SIZE=1000 ACTION=REJECT NDN-INFO="Message too large"
```

Messages over 1Mbyte in size are rejected, and a Non-Delivery Notification containing the text "Message too large" is returned to the message originator.

Notifying Deferred Message Delivery

```
PRIORITY=LOW ACTION=DEFER DAY=5 TIME=09:00-18:00 NOTIFY="!delaynote"
```

The delivery of any low priority messages is deferred between 9 a.m. and 6 p.m. on Friday. The following text (contained in the file ~/rules/delaynote) is appended as a text part before the standard notification text, which is routed to the message originator without delay according to the normal Service Router routing process:

```
Item 1 (distribution list)
.
.
.

Item 2 (delaynote)
To speed the delivery of normal and urgent messages during normal office
hours (9am-6pm) on Fridays, any low priority messages are not delivered
during these hours.

Item 3 (standard notification text)
Delivery of your message has been delayed for some recipients:
```

```
------- Deferred Mail Manager Report from 'machine_x' -------
Message Reference:   S3086127
Message Id. String:  H000003d00015469
Routing rule name:   Low priority messages sent at 6p.m.
Routing rule action: DEFER
Deferred until:      Fri Apr 12th 18:00:01 1997
Message Priority:    NON-URGENT
Recipients affected:

Amy Flinders / SPE, Sales
John Person / unix (johnp@acmesoft.com)
Frank Underwood / SPE, Training
```

Deferring Delivery of a Batch of Messages

```
PRIORITY=LOW ACTION=DEFER TIME=8:00@1:00
```

Any low priority message received from 8 a.m. are held in a batch and delivered at 1-hour intervals, that is, at 9 a.m., 10 a.m., 11 a.m., and so on.

Example Script for SUBJECT Attribute

Below is an example of a script that can be used with the SUBJECT tag. This script (/opt/scalix/examples/general/subject) is supplied with Scalix. You can copy this script into ~/rules and modify it as required. You must use the protocol syntax as defined in the comments at the top of the supplied script. You must also ensure that the script is both readable and executable.

```
#!/bin/sh

#####################################################################
#Scalix Router Subject Mapping Protocols
#####################################################################
# PROTOCOLS SYNTAX:
# The following table outlines the possible commands sent by
# Scalix and the expected replies sent by the Mapper. Note:
# 1) each command/reply must end with a new line (n) character
# 2) the Mapper must NOT buffer its output, each reply must be
#    flushed
# 3) the Mapper must reply to each command
# COMMAND          REPLY            REPLY COMMENTS
# ================================================
# <start>          220<SP><text>    Mapper must output this when
#                                   starts up
# HELO<SP><text>   250<SP><text>    Mapper accepts Scalix session
# SUBJECT:<text>   251<SP><text>    Subject does not match requirement
# SUBJECT:<text>   252<SP><text>   Subject matches requirement
# QUIT<SP><text>   221<SP><text>   Mapper terminates session
# <others>         500<SP><text>   Unexpected command/syntax
#####################################################################
# handle "<start>"
# return ready status
rep="220 Subject Mapper Ready"
echo "$rep"
```

```
# loop to process commands
Quit="FALSE"
while read cmd
do
     case "$cmd" in
   "HELO"*)
         # handle "HELO<SP><text>"
         # return ok status
         rep="250 Ok"
         ;;
   "SUBJECT:PUBLIC:"*)
         # handle "SUBJECT:<text>"
         # subject matches requirement, strip off "SUBJECT:"
         subject=‘echo $cmd | sed -e "s/SUBJECT://"‘
         rep="252 $subject"
          ;;
   "SUBJECT:"*)
         # handle "SUBJECT:<text>"
         # subject does not match requirement, return reason
            rep="251 Subject Does Not Match"
          ;;
   "QUIT"*)
          # handle "QUIT<SP><text>"
          # return status, set flag to exit loop
           rep="221 Subject Mapper Close"; Quit="TRUE"
          ;;
   *)
           # handle "<others>"
           # return error status
           rep="500 Unrecognized Command or Syntax Error"
           ;;
   esac

   # must reply to each command
   echo "$rep"
   if [ "X$Quit" != "XTRUE" ]
   then
    continue
   else
      break
   fi
done
exit 0

#############################################################
# End of script
#################################################################
```

Deferred Mail Manager

The Service Router automatically starts the Deferred Mail Manager process (defer.manager), which handles any messages whose delivery is to be deferred.

The Deferred Mail Manager monitors the DMM queue, and picks up any new messages submitted to it by the Service Router when all the attributes of a rule are matched and ACTION=DEFER is specified.

All deferred messages and the deferral period defined in the specified ruleset are stored in the deferred message list (~/msglists/DEFER.SR).

When the deferral period specified in the rule is reached, the Deferred Mail Manager delivers the deferred messages. The Deferred Mail Manager does not resubmit messages to the Service Router, but routes the messages itself.

Listing Deferred Messages

To view a list of deferred messages, enter this command:

```
omstat -d
```

You can attempt to force the delivery of a deferred message by using the omresub command. For details of these commands, see the online manual entries.

Loop Detection

The Service Router uses the following three methods to prevent a message "looping" around the system:

- Transport trace record
- Hop count indicator
- Redirection history record

All messages contain the items listed above and each of these records/indicators is checked by the Service Router as it processes a message.

Transport Trace Record

A transport trace record is added to each message when it is routed by the Service Router. The transport trace record contains the host name of the machine and the primary mailnode for that system.

When the Service Router processes a message, it checks for the transport trace record. If a transport trace record is found that has the same host name and primary mailnode as the current system, the message has looped.

If the message has looped and the matching transport trace record is followed by an "operation trace" record (such as "recipient redirected" added by the Local Delivery Service), the loop is allowed. If the matching transport trace record is not followed by an operation trace record, then all processing of the message is discarded, and a Non-Delivery Report is generated for each address in the active recipient list. The Non-Delivery Reports are then returned to the originator.

Hop Count Indicator

Each message contains a "hop count". The hop count is incriminated with each hop (delivery service to Service Router) that the message makes. If the hop count exceeds 100, then the message is considered to have looped.

If the message has looped, then all processing of the message is discarded, and a Non-Delivery Report is generated for each address in the active recipient list. The Non-Delivery Reports are then returned to the originator.

This trap is mainly of use in messages that have arrived from non-Scalix systems (such as the internet) that use a hop count to detect loops.

Redirection History Record

Each recipient in the active recipient list can carry redirection history. The redirection history lists the previous redirections and address resolutions that have been performed for the recipient.

When a message is redirected, the Service Router checks the redirection history to make sure the message is not being sent back to a previous redirection. If the Service Router detects a loop (the message is going to be sent back to a previous redirection), the Service Router generates a Non-Delivery Report and sends it to the originator instead of redirecting the message again.

Routing Table Commands

The following table details the commands associated with the Routing Table.

Command	Description
omaddrt	Add a route
omdelrt	Delete a route
ommodrt	Modify a route
omshowrt	List routes and show how an address is routed

Chapter 7

Scalix Directories

This chapter provides detailed information about Scalix Directories, directory structure, and the commands you use to manage them. This chapter includes the following information:

Directory Overview

Scalix services, gateways, and clients use Directories to resolve incomplete addresses. Clients can also use Directories to obtain additional information such as the telephone number, job title, or the postal address of a user.

Note	When the term "Directory" is used with an initial uppercase letter, it refers to a Scalix lookup Directory. When the term "directory" is used with a lowercase letter, it refers to a directory in the Linux file system.

A Directory comprise a series of *entries*. Each Directory entry identifies a user (or entity) and consists of attributes that belong to that user.

There are two types of Directories in Scalix:

- Shared Directories - these can be accessed by a number of users.
- Personal Directories - these can only be used by a specific user.

Scalix must have a default Shared Directory. This is the Directory used by the Service Router when it resolves addresses (see "The Service Router" on page 69) and contains the master list of all users in the network. You can add other shared and personal Directories to Scalix to fulfill the requirements of specific users or groups of users. Access to shared Directories is controlled through the use of Access Control Lists (see "Access Control Lists" on page 197).

The attributes that make up Directory entries are defined in the attribute definition file. This file contains the internal attribute tags, syntaxes, and lengths of all attributes defined for use on the Scalix system. You can customize Scalix directories by adding fields and that can be automatically generated each time you add an entry to the directory.

The internal attribute tags defined in the attribute definition file are mapped to language dependent tags and descriptions in the localized attribute file. The localized attribute file provides local language display of attribute tags and descriptions.

There can be any number of Scalix Directories. Each Directory is held in a database structure in a separate directory under ~/dir. Scalix provides a number of diagnostic utilities for monitoring the use and integrity of Directory databases.

The Scalix directory system is not only used for the storage and retrieval of names and addresses, but also contains Public Distribution Lists (PDLs). You can manage access to PDLs using Access Control Lists. Also, using the Scalix LDAP server, you can access a Scalix directory using the LDAP protocol.

Directory Structure and Functions

One of the functions of the Scalix directory system is the storage and retrieval of names and addresses. This function is shared across the following directories:

- The SYSTEM directory is the default directory for name and address storage and retrieval.
- The USERLIST directory is used by services (especially the Local Delivery service) to verify local mailnode and Scalix recipient information.
- The FREEBUSY directory is used to allow MAPI users to share calendar information. You must manually create this directory.

The SYSTEM Directory

The following components access the SYSTEM directory:

- The Service Router accesses the SYSTEM directory to verify recipient mailnodes if it additional information is required by the Service Router to route the message.

  For example, if the Service Router receives messages addressed to a mailnode for which it has no routing information, the Service Router accesses the SYSTEM directory to locate the missing routing information. The Service Router then tries to route the message again using the address found in the directory.

  If the Service Router directory search fails (either because the routing information is not present or their are too many possible routes), a Non-Delivery Notification (NDN) occurs.
- Clients access the SYSTEM directory (and Message Store) using the User Agent Layer (UAL). When a user enters a name in the To field, the client requests a SYSTEM directory search and returns a list of possible names.

Client directory searches include:

- completion of a partial address entered by a user
- substitution of real addresses for nicknames
- identification user addresses from, for example, their respective job title.

  However, the Outlook client requires directory entries to be sorted in a particular manner. The Client Directory Access (CDA) server performs the sorting of directory entries. See “Client Directory Access” on page 98. for more information.

Scalix also includes an LDAP server which allows you to perform LDAP searches on a Scalix directory from the Outlook client. See “About the POP3 Server” on page 187 for more information.

- The Internet Mail Gateway accesses the SYSTEM directory when it needs to convert addresses. If the Internet Mail Gateway receives a message where the recipient address does not contain a DDA/Foreign address, the Internet Mail Gateway searches for the DDA in the recipients directory entry.

The USERLIST Directory

This directory contains information about all the local addresses and valid recipients on the Scalix Server. Each time you add a mailnode or local recipient, an entry is automatically added in the USERLIST directory. A user entry includes items such as name, mailnode, Scalix user ID, Linux user ID, privileges, and Scalix password (including when the password was last changed).

The Local Delivery service always accesses the USERLIST directory to verify local recipient information. If the Local Delivery service cannot find a matching entry in the USERLIST directory, a NDN is generated.

You can access the USERLIST directory like other Scalix directories. Because the USERLIST directory is “hidden”, you must specify -t h (type=hidden) if you want to search the directory.

```
omsearch -d USERLIST -t h -e s=*
```

Caution	Scalix recommends that you do not manually edit the USERLIST directory. Use commands such as ommodu and ommodmn to modify recipient and mailnode information. However, this directory is hidden (the omlistdirs command does not display the existence of this directory).

FREEBUSY Directory

The FREEBUSY directory is used to allow MAPI users to share calendar information. This directory is created automatically when you install Scalix. The local calendar information of each user is periodically added to the shared FREEBUSY directory on the local Scalix Server. All users who want to share calendar information need to have an entry in this directory. Users are added automatically when you use the -f option in the omaddu or ommodu commands.

When a user attempts to schedule an event involving more than one user in the client calendar application, the Scalix Directory Relay service queries the appropriate FREEBUSY directories on the Scalix Server to determine the availability of users and, if necessary, other resources such as facilities.

Directory Entries

A Directory contains addressing information (including the Internet Address) and can store additional information such as a telephone number, office location, and company name.

You can add customized attributes and these are entered or displayed using a TAG=value pair. The TAG identifies the type of attribute, and implicitly defines the syntax and size of the value that the attribute can have.

The dir.attribs file defines attribute types and the diratt.loc file (one for each language installed on the Scalix server) provides language dependent tags and descriptions for each attribute type. New attribute types are defined by editing these files. You can display attribute types available on Scalix by entering the omshowatt command. See (ADDRESSING) for more information about Directory entry attributes.

Directory entries are added, deleted, and modified using the omaddent,omdelent, and ommodent commands. See “Using Directory Commands” on page 96 for more information about Directory commands.

Using Directory Commands

Use the following commands to manage Scalix directories:

Listing Fields In The SYSTEM Directory

To view the fields defined for the SYSTEM directory, enter this command:

```
omshowatt
```

The following information appears, with the output sorted by Field Name, Data Type, Length, and Descriptive Name—as illustrated here.

```
S                   KX          40          X.400 Surname
C                   X           3           X.400 Country Code
DIT-RDN                         U/L         DIT Relative Distinguished Name
DIT-RDN-ID          KPS         40          DIT RDN Global Identifier
DIT-PARENT-ID       KV          40          DIT DN Parent Identifier
DIT-OWNER           V           U/L         DIT node owners
DIT-SEARCH-REF      V           U/L         DIT search referrals
DIT-MODIFY-REF                  U/L         DIT modify referral
FULL-NAME                       64          Personal Full Name
LAST-NAME           K           64          Personal Last Name
MIDDLE-NAME         V           20          Personal Middle Name
FIRST-NAME          K           20          Personal First Name
ALIAS               KMSV        20          Personal Aliases
...
```

Creating New Directories

To create a new directory, enter:

```
omnewdir -d directory name -t
```

Adding the p parameter at the end of the command creates a “private” directory that is only accessible by the user who creates the directory.

New directories that you create have default attributes.

Adding and Modifying Directory Entries

To add a directory entry to the SYSTEM Directory, enter the following:

```
omaddent -e "s=lastname/g=firstname/ou1=mailnode of unix gateway/
ou2=value/ou3=value/cn=display name/ia=user@domain.com"
```

To modify an entry (for example, with a new field named Job Title with an associated value):

```
ommodent -e s=lastname -n Job Title="Accountant"
```

Searching Directories

An entry in a directory is retrieved using a search filter. The filter is compared with the entries in the directory. All entries in the directory that match the filter are returned.

Us the omsearch command to search a directory. For example, the following command returns all entries in the system default directory containing a surname of Smith:

```
omsearch -e S=smith
```

A filter is made up of "filter items" and other filters. A filter item is an attribute tag, an attribute value, and a matching operator. For example, S=ExampleEntry. The following table lists available matching operators:

Operator	Description
=	Equal to the value of the filter item.
-	Approximately equal to the value of the filter item (supported with filter items using ASCII string type syntaxes).
>	Greater than or equal to the value of the filter item (supported with filter items using integer type syntaxes; for example INTEGER and DATE).
<	Less than or equal to the value of the filter item (supported with filter items using integer type syntaxes; for example INTEGER and DATE).
&	AND.
|	OR.
!	NOT.

Use parentheses ( ) to nest filters within other filters. For example, to search the system default directory to find all entries that have a surname of “Wolf”, and a given name of “Mike” or “Mikey” but not “OU1=sales”, the following filter could be used:

```
omsearch -e "S=Wolf & (G=Mike | G=Mikey) &! OU1=sales"
```

Wildcards (*) can also be used in filter items with string type syntaxes to represent whole or partial attribute values (this applies to string syntaxes only).

To list all surnames, enter:

```
omsearch -e s=*
```

Or enter the following to search for entries containing a specific string of characters:

```
omsearch -e s=*wo*/ou1=*sal*
```

To display only a specific attribute value, use the -m parameter. Enter:

```
omsearch -e s=* -m OU1=
```

Only the OU1=Sales attribute displays for all the surnames returned by omsearch.

Directory Command Summary

The following table lists commands associated with Scalix Directories.

Command	Description
omaddent	Add one or more entries to a Directory
omdelent	Delete one or more entries from a Directory
omdiropt	Optimize a Directory
omdoptall	Optimize all Directories
omfmtent	Format Directory and address attributes
omlistdirs	List Directories
ommoddir	Modify a Directory
ommodent	Modify a Directory entry
omremdir	Delete a Directory
omsearch	Search a Directory
omshowatt	Show available attribute types

Client Directory Access

The Client Directory Access (CDA) Server builds access tables for Scalix Directories to provide sorted lists of Directory entries.

The Outlook client used with Scalix requires sorted entries in the Address Books. This enables “typedown” functionality when selecting addresses in the interface. The CDA Server is used to provide the sorted lists of Directory entries.

The commands omaddcda, ommodcda, omdelcda, and omshowcda, add, modify, delete, and show Directories configured for processing by the CDA Server, respectively. The omexeccda command forces the immediate processing of a Directory without waiting for the next periodic rebuild of its access tables.

Options you set in the general.cfg file determine how the CDA Server operates. See “Configuration Options” on page 301 for more information.

Using the CDA Server

To start the CDA Server, enter:

```
omon -s cda
```

The CDA Server periodically checks its configuration settings (by default, once every 5 minutes in ~scalix/sys/cda.cfg) and if the processing of a Directory is required, the CDA accesses the Directory, extracts the required information, sorts the entries (on surname, given name, and initials by default), and stores the entries in access tables within the ~/cda directory.

As the access tables for a Directory are created periodically, modifications to a Directory are not immediately reflected in the access tables. Changes to Directory entries become visible to a client as follows:

- Added entry: The new entry is visible only after the CDA Server (or omexeccda) rebuilds the access tables and the client closes and opens the Directory.
- Deleted entry: The substitute text <Deleted Entry> appears until the CDA Server (or omexeccda) rebuilds the access tables and the client closes and opens the Directory. Then neither the entry nor the substitute text appears in the Directory.
- Modified entry: The change is immediately visible, but the sort order can be incorrect until the CDA Server (or omexeccda) rebuilds the access tables and the client closes and opens the Directory.

To force the server to process a directory immediately, you can use the omexeccda command.

You can configure the time interval between the processing of a Directory by the CDA Server. The interval is configured by the omaddcda and ommodcda commands. By default, the interval is 24 hours. The practical minimum interval depends on the time taken for the CDA Server to build the access tables. This in turn depends on a number of factors such as system size, system resources, system loading, and Directory size. If the interval is set too low, the CDA Server might continuously processes the Directory. To verify the amount of time a Directory takes to process, use the command omshowcda -d Dir_name.

To optimize the rebuilding of the access tables, configure the CDA Server to check the Directory change log. Do this by setting the option CDA_USE_CHANGE_LOG=TRUE in the general.cfg file See “Configuration Options” on page 301 for more information.

For the Outlook client user, Public Distribution List Directory entries might appear not to exist if the user does not have the required privileges defined by the Access Control List (ACL) for the PDL. For example, when accessing the Address Book, any PDL for which the user does not have read privileges is replaced by the text “<Deleted Entry>”.

CDA Command Summary

The following table lists commands associated with the CDA.

Command	Description
omaddcda	Add a Directory to the CDA Server configuration. You can specify the directories to be processed, configure the interval at which the directories should be reprocessed, the fields to be used to sort directory entries, and how often the CDA server re-reads its configuration details.
omdelcda	Delete a Directory from the CDA Server configuration.
omexeccda	Force the CDA Server to process a Directory immediately.
ommodcda	Modify the CDA Server configuration for a Directory.
omshowcda	Show the CDA Server configuration for a Directory.

Chapter 8

Integrating Scalix with Microsoft Active Directory

This chapter covers the following topics:

Introduction

If you want to manage some or all of your Scalix accounts (users and groups) with Microsoft Active Directory, you can do so after completing a series of tasks detailed here.

Note	If you have existing user and group records that you manage through Scalix utilities, you can create separate records on Scalix for Active Directory management. Note that with two separate collections, you must use the appropriate tool to manage the records—Scalix Administrative Console for non-Active Directory records, and ADUC for Active Directory-specific records. (You can open and read the contents of Active Directory records with SAC, but you cannot modify or delete anything.

Overview of the setup tasks

- Install and run Scalix *ForestPrep,* to add schema extensions to Active Directory.
- Create and test an omldapsync agreement between Scalix and Active Directory, then schedule a regularly-occurring synchronization of Active Directory records to your Scalix system.
- Install the Scalix Active Directory GUI extensions on every administrative workstation running Microsoft *Active Directory Users and Computers (ADUC)*.

  At this point you can start **populating** Scalix-specific attributes inside Active Directory.
- [Optional] Activate authentication between Scalix, Active Directory and your Kerberos-based security system.

Each of these tasks is detailed separately in the following sections. Once these are finished, you can use Active Directory to create and manage all your Scalix users—also detailed in this guide.

To begin, you will want to install Scalix extensions to the Active Directory schema, as described in the next section.

1: Installing the Scalix Schema Extensions in Active Directory

Installing and running Scalix *ForestPrep* extends the Active Directory schema with new Scalix-specific object classes and attributes. (See “A Complete List of the Scalix Extensions of Active Directory” on page 121 for a complete description of these extensions.)

Alert	Remember that adding extensions to Active Directory is irreversible.

Installing Scalix ForestPrep

1. Log in to the host on which the schema master is stored, using an administrator account with schema administrator rights. Or as an alternative, log in to a workstation with access to the schema master host. (Note: Scalix ForestPrep automatically detects the schema master when it starts.)

2 Launch the *Scalix Active Directory Extensions* installer.

3 Work through this installation wizard (in which you install the utility).

When installation is complete, Scalix ForestPrep is located in this directory:

```
c:\Program Files\Scalix\Administration\
```

Running Scalix ForestPrep

This process adds a set of Scalix-specific extensions to Active Directory, allowing you to remotely manage your Scalix-based users and groups with Active Directory.

1 Open a command prompt window.

2 Navigate to this directory:

```
c:\Program Files\Scalix\Administration\
```

3 Run this application:

```
Scalix AD Extensions.msi
```

If the installation is successful, an "update successful" message appears in the window.

4 Exit the terminal window.

Note	Errors, if any, are logged in to the Event Viewer, providing you with a permanent record in case of Scalix/Active Directory problems that you suspect are related to the extensions.

After you install the Scalix Active Directory extensions

After you install the schema extensions, an automatic five-minute waiting period is enforced by Active Directory. This ensures that all additions or changes do not upset current processes. After the mandatory five minutes, the Scalix extensions activate. The system can now run the updated ADUC clients (see following) and use Active Directory to manage Scalix-hosted mail users and groups.

Additional information about Scalix ForestPrep and the Active Directory extensions

If you accidentally re-run ForestPrep, no problems will arise, as ForestPrep detects the GUID from any previous installation, and won't overwrite the existing extensions.

If you are unable to complete the installation, just re-run ForestPrep, which picks up where it left off and finishes the installation.

Alert	Active Directory may take a long time to disseminate the new Scalix extensions through the system. Key factors include the number of domain controllers, the number of Active Directory servers and connection speeds between network resources. Additionally, the older the Windows OS underneath AD, the slower the full update will be; Windows 2000-based systems will require a complete Active Directory database resynchronization while Windows 2003-based systems will take less time to propagate changes. Your particular Active Directory system may be updated in minutes—or may take a weekend.

2: Installing the ADUC GUI extensions

You have several options for updating all your users' and computer workstations with Scalix-specific Active Directory GUI enhancements:

- Distribute the Scalix installer for individual per-station installation via floppy drive, network-accessible file sharing or Web FTP.
- Use a third-party utility to script a mass installation that installs the extensions when an administrator logs in to the Active Directory server.
- Use Active Directory itself (via GPO) to propagate the users' and computers' GUI enhancements.

The Scalix installer should only be run on ADUC workstations. After installation, Active Directory now includes a *Scalix Server* tab in the [user] and [groups] Properties dialog boxes, with options relevant for users or groups who are on the Scalix system.

1. For a first-time installation of the Scalix GUI enhancements on an ADUC workstation, you must log in using a Windows administrator with local admin rights.
2. If they are not already present, copy the files "Scalix AD GUI Extensions.msi" and "Scalix AD Schema Extensions.msi" to the workstation desktop.
3. On the Active Directory master server, install the schema package by launching "Scalix AD Schema Extensions.msi." (This is the machine where you ran ForestPrep.) This is an irreversible, one-time-action that updates your Active Directory schema.
4. On every admin workstation and every server with the local ADUC GUI, install the GUI package by launching "Scalix AD GUI Extensions.msi."
5. Work through the wizard for each installation process.
6. Click **Finish** when each process is complete.

   ADUC is now ready for Scalix account management.

3: Setting Up the Synchronization Agreement

Before omldapsync (a command set on the Scalix server) can communicate with the Active Directory server (to download AD-managed user and group account information) a custom synchronization agreement must be created and configured. Once this agreement has been tested successfully and run at least once, you can implement a cron job to automatically run omldapsync (and this agreement) on a regular basis.

Requirements

- Log in to the Scalix server as root.
- Have the domain name of Active Directory and Scalix servers.
- Have the URL of the Scalix server running Scalix Administration Server (if you have more than one server in your Scalix system.
- Have the authentication ID and related password for the Scalix administrator.
- Have the administrator ID and related password for the Windows/AD server.

Procedure

To prepare and test a new omldapsync agreement, follow these steps:

1. Log into the Scalix Server as root.
2. To run omldapsync in "interactive" mode, enter this command at the prompt:

   ```
   omldapsync -i [syncid]
   ```

   - Replace "**[syncid]"** (a placeholder) with a unique name for your Active Directory-Scalix synchronization agreement. The name should be no more than six alphanumeric characters in length; for example, **AD_SX1**.

   After you press Enter, the omldapsync "common tasks menu" appears, followed by a numbered list of interactive setup/administrative tasks.
3. Enter "1" (one) at the prompt, and press Enter.

   Omldapsync creates the subdirectory for the newly named synchronization agreement along with the [agreement_name].cfg file.
4. At the next prompt, you'll be asked to select the synchronization agreement type.

   ```
   Select sync agreement type to create (00):
   ```
5. Enter "11" (eleven) at the prompt and press Enter.

   The first of a series of interactive configuration prompts now appears:

   ```
   INPUT: value for SCALIXHIDEUSERENTRY (scalixHideUserEntry):
   ```
6. Press Enter to accept the default value for this prompt and for all of the following value queries, listed below:

   ```
   INPUT: value for SCALIXHIDEUSERENTRY (scalixHideUserEntry):

   INPUT: value for SCALIXMAILBOXCLASS (scalixMailboxClass):

   INPUT: value for SCALIXLIMITMAILBOXSIZE (scalixLimitMailboxSize):
   ```

```
INPUT: value for SCALIXLIMITOUTBOUNDMAIL (scalixLimitOutbound-
Mail):

INPUT: value for SCALIXLIMITINBOUNDMAIL (scalixLimitInboundMail):

INPUT: value for SCALIXLIMITNOTIFYUSER (scalixLimitNotifyUser):

INPUT: value for EX_SCALIX_MAILBOX (scalixScalixObject):

INPUT: value for EX_SCALIX_MAILNODE (scalixMailnode):

INPUT: value for EX_SCALIX_MSGLANG (scalixServerLanguage):

INPUT: value for EX_SCALIX_ADMIN (scalixAdministrator):

INPUT: value for EX_SCALIX_MBOXADMIN (scalixMailboxAdministrator):
```

7 When this prompt appears:

```
Edit config file now y/n (n):
```

Press “Y” for Yes.

8 When this prompt appears:

```
Use vi to edit y/n (n):
```

9 Press “N” to be guided through an interactive session, in which you can efficiently enter the configuration settings. (The option is to press “Y”, and use VI to edit the configuration file manually—which is not documented in this guide.)

The rest of this procedure details the interactive sequence of queries.

10 The first configuration prompt (JAVA_HOME) asks for the location of the Java installation on the Scalix server.

Enter the full pathway for the Java directory.

11 The next prompt (EX_HOST) asks for the remote LDAP server name.

Enter the name of your Active Directory server.

12 The next prompt (EX_LOGON) asks for the Active Directory administrator account name. The format for your entry should be:

```
cn=administrator,dc=organization,dc=com
```

13 The next prompt (EX_PASS) asks for the related Active Directory Admin password. Be sure to enter “1” (one), so that the synchronization can be fully automated.

14 The next prompt (IM_HOST) asks for the fully qualified domain name (FQDN) of the Scalix server on which the directory will be stored. If you have one server, enter that domain name. If you have several servers in your Scalix system, enter the FQDN of the server on which Scalix Administration Server is running.

The format should be

```
server_name.domain.com
```

15 The next prompt (IM_CAA_URL) asks for the URL of the Scalix server on which Administration Server is running. If you have one server, enter that URL.

The format should be:

```
http://local_server.domain.com:8080/caa/
```

16 Be sure to end the URL in a slash, as shown above.

Note	If you are setting up synchronization on a Scalix server running v10 of Scalix, enter a URL without the 8080 port number: http://local_server.domain.com/caa/

17 When the next prompt (IM_CAA_KEYSTORE) appears, press Enter to accept the default of no entry.

18 When the next prompt (IM_CAA_ID) appears, enter the authentication ID for a full Scalix administrator. The authentication ID is separate from the administrator's mailing address or display name.

19 When the next prompt (IM_CAA_PASS) appears, type the password associated with the Scalix administrator authentication ID.

Note	Ideally, you will already have verified the usability of this authentication ID and password by logging into Scalix with Scalix Administrative Console using this administrator account.

20 When the next prompt appears (EX_BASEn) [*with "n" being replaced by a number*], enter the container name and its full LDAP suffix, as shown here:

```
EX_BASE1:cn=users,dc=scalix,dc=com
```

- If needed, you can list up to nine sequentially numbered containers at this time, if used for Scalix users and groups on Active Directory.

21 When the next prompt (EX_SCALIX_MAILNODE) appears, enter the mail node in this format:

```
EX_SCALIX_MAILNODE=scalixMailNode
```

This query completes your omldapsync synchronization configuration. You'll now proceed through testing and use of the omldapsync agreement.

22 When this prompt appears:

```
Compare old config to new y/n (?):
```

Type "Y" for Yes.

omldapsync displays a summary of this new configuration on-screen.

23 When this prompt appears:

```
Replace old config with new y/n (?):
```

Type "Y" for Yes.

A series of status messages now appear, noting that the updated file was "installed".

24 When this prompt appears:

```
Attempt to test data extraction now y/n (n):
```

Type "Y" for Yes.

Omldapsync now initiates a non-destructive test of the synchronization communication parameters. No user data will be downloaded from Active Directory to Scalix at this time.

- A series of status messages appears, as omldapsync contacts both servers and establishes the connection.

25 If the test is successful, this message appears:

```
[DATE TIME] STATUS: Configuration of [AGREEMENT_NAME] completed
```

26 If the test fails, you will want to edit the configuration file to correct the problem entry, then re-test the data extraction.

The "configuration completed" message is followed by the omldapsync interactive menu.

27 Press "2" (number two) to start loading all the Active Directory-specific users in a Scalix directory.

- After the synchronization is initiated, a series of status messages report the success of various synchronization actions: new users added, users deleted, new limits applied, etc. You should review this list for the "entries failed" counts in each category.

28 If the download is unsuccessful, you may see a direction to a log file, a SOAP failure report with details, or a prompt to run an omldap utility that will help you fix the problems—after which you can re-start the users download again.

29 When the loading is complete, another series of status messages concludes with:

```
LDAP dir sync export [AGREEMENT_NAME] completed
```

If the synchronization is successful, your Scalix server now hosts a set of users and groups managed by Active Directory.

30 You should now set up a cron job to run this omldapsync agreement at the regular time intervals of your choosing.

Alert! This newly configured Active Directory/Scalix synchronization is uni-directional; Active Directory records are downloaded to Scalix. This means that you can use Scalix utilities to fully manage Scalix-generated user and group records, but you should only use Active Directory to manage all your Active Directory-generated/controlled records. Changes made with other utilities will be erased in the next synchronization.

Manually running omldapsync

1 To manually run this agreement at any time, log into Scalix, then enter this command:

```
omldapsync -u [AGREEMENT_NAME]
```

2 The Active Directory directory will be downloaded to Scalix, and when finished, a series of status messages will end with this line:

```
LDAP dir sync export [AGREEMENT_NAME] completed
```

Managing Scalix Users and Groups with Active Directory

You can now use *Active Directory Users and Computers* (ADUC) to create new Scalix user or group accounts, to modify existing accounts, or (if needed) to delete the accounts. Remember that deletion is final, and totally erases record and associated data.

Alert!	You should know that you can use Scalix CLI to open and change Active Directory-specific records on the Scalix server, but any changes you make will be over-written in the next Active Directory/Scalix synchronization. Remember, you can use Scalix utilities to fully manage Scalix-generated user and group records, but you should use Active Directory to manage all your Active Directory-generated/controlled records.

After detailing the addition of Scalix mailnode information to Active Directory, this chapter covers the two main Active Directory/Scalix account administrative tasks:

- Managing a user account
- Managing a group account

Each is described separately in the following sections.

Adding Scalix Mailnode information to Active Directory

After you've updated the Active Directory schema and installed GUI extensions for ADUC on every relevant workstation, you need to add Scalix mail node information to the "scalix" directory on the AD server. This data will be accessible in the new Scalix Server tab in the [user] Properties dialog box.

To create and store a file containing mailnode information for use in AD, follow these steps:

1 Use Notepad or Wordpad to create and save a new text file with this name:

```
mailnodes.txt
```

2 In the body of this file, type each mailnode as a separate entry.

3 Save the text entries.

4 Copy this file to this directory on the AD server:

```
c:\Program Files\Scalix\Administration\
```

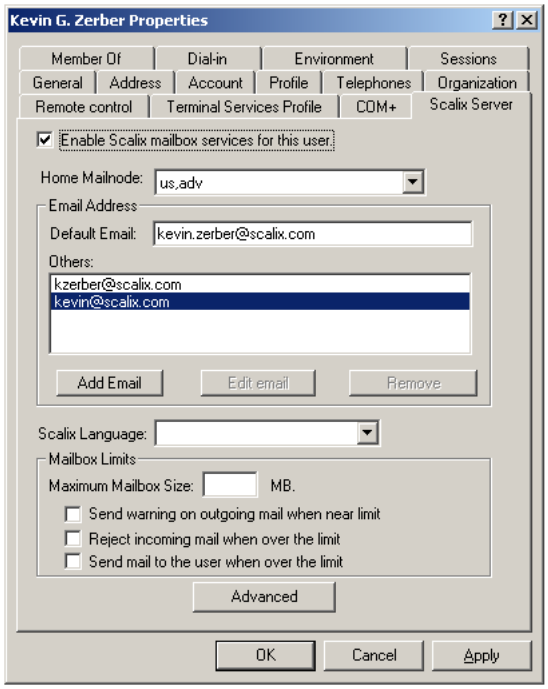
The mailnode information is ready for use in Active Directory.

Creating a new Scalix account in Active Directory

1 If you have not already done so, start ADUC.

2 Follow the steps to create a new user or group, that will be associated with a Scalix mailbox.

3 In the [user name] **Properties** dialog box, click the **Scalix Server** tab (shown below).



4 You can enter the following required information:

- Check **Enable Scalix Mailbox Services...** (if no checkmark is present).
- Open the **Home MailNode** menu and choose the appropriate Scalix mail node (if additional options are available) that this account is associated with. Otherwise the server default will be the only choice.
- Verify the address in the **Default E-mail Address** fields.

5 Enter the following optional information:

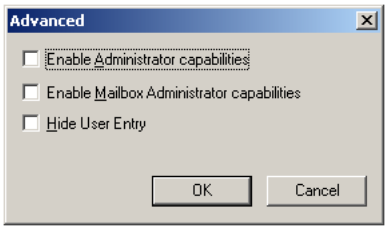
- **NOTE:** This dialog box now includes a “Friendly name” (Display Name) field, followed by a mailbox name field and the domain field. The previous screen shot does not represent these options, but they are present.
- Click **Add E-mail** if there are other e-mail addresses associated with this account. A dialog box appears, in which you can type the address. (You can sort the addresses in this list, if more than one is listed.)
- Open the **Scalix Language** menu and choose the primary language in use on this server.
- Use the **Mailbox Limits** to set a maximum capacity (if the system limit is not appropriate) and activate various user-alert options.

Note	The Default E-mail fields in both the General tab and Scalix Server tab are linked. Change one entry and the other will automatically update-- if you click **Apply**. If you change one, then open the other and change that without clicking Apply, problems may occur.

6 Click **Apply** after making any changes.

7 Click **Advanced**.

The Advanced dialog box appears.



- Click **Enable Administrator Capabilities** to permit this user to act as a "full administrator". (See *Managing a Scalix System with the Administrative Console* for more information.)
- Click **Enable Mailbox Administrator Capabilities** to permit this user "Scalix User Information" permission, to open other user accounts and modify personal settings. (See *Managing a Scalix System with the Administrative Console* for more information.)
- Click **Hide User Entry** to prevent this account record from being visible in the public directory.

8 Click OK to save your entries and close this dialog box.

9 Click **Apply**, then click **OK** to save your entries and close the [user name] Properties dialog box.

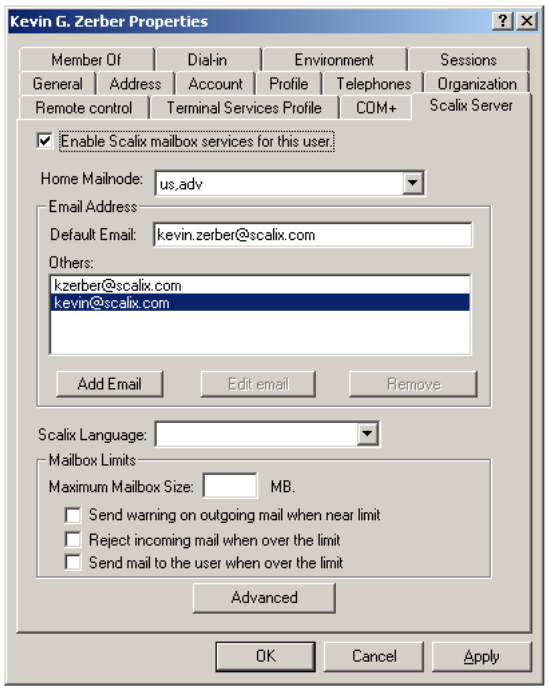
After the next omldapsync operation, this mailbox will be in effect.

Managing a user account

1 Start ADUC.

2 Locate a user record you want to review.

3 Open that record's [user name] **Properties** dialog box.



4 Click the **Scalix Server** tab (shown above).

5 Enter or modify the following required settings:

- Check **Enable Scalix Mailbox Services...** (if no checkmark is present).
- Open the **Home MailNode** menu and choose the mail node (if other options are available) that this account should be associated with—if a change has been made.
- If the address in the **Default E-mail Address** field needs alterations, make them at this time.

6 Enter or modify the following optional settings:

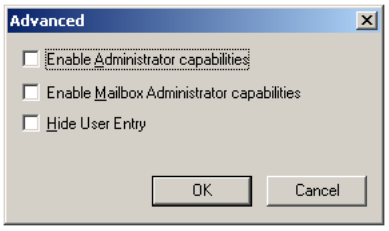
- **NOTE:** This dialog box includes a "Friendly name" (Display Name) field, followed by a mailbox name field and the domain field. The previous screen shot does not represent these options, but they are present.
- Click **Add E-mail** if there are other e-mail addresses associated with this account. A dialog box appears, in which you can type the address. (You can sort the addresses in this list, if more than one is listed.)
- Open the **Scalix Language** menu and choose the primary language in use on this server.
- Use the **Mailbox Limits** to set a maximum capacity and activate various user-alert options.

Note	The Default E-mail fields in both the General tab and Scalix Server tab are linked. Change one entry and the other will automatically update-- if you click **Apply**. If you change one, then open the other and change that without clicking Apply, problems may occur.

7 Click **Apply** after making any changes to these settings.

8 Click **Advanced**.

The Advanced dialog box appears.



- Click **Enable Administrator Capabilities** to permit this user to act as a "full administrator". (See *Managing a Scalix System with the Administrative Console* for more information.)
- Click **Enable Mailbox Administrator Capabilities** to permit this user "Scalix User Information" permission, to open other user accounts and modify personal settings. (See *Managing a Scalix System with the Administrative Console* for more information.)
- Click **Hide User Entry** to prevent this account record from being visible in the public directory.

9 Click OK to save your entries and close this dialog box.

Alert	DO NOT CLEAR the "Enable" checkmark unless you are sure you want to delete the mailbox along with all user data. That record will nominally remain in Active Directory, but there will be no data or mail.

10 Click **Apply**, then click **OK** to save your entries and close the [user name] Properties dialog box.

After the next omldapsync operation, the changes will be in effect.

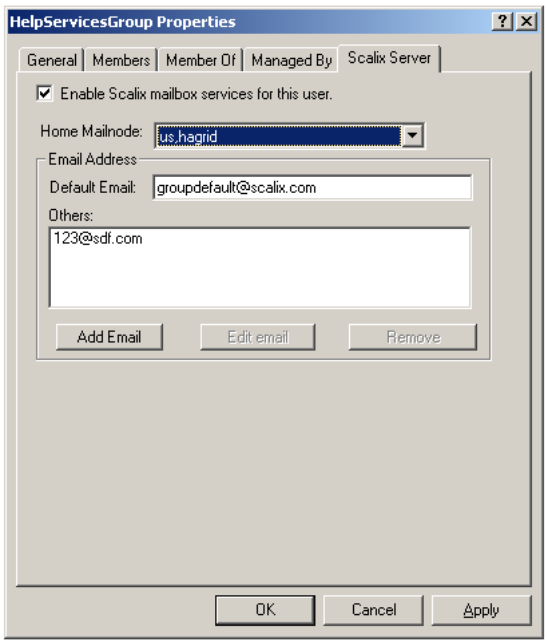
Managing a group record

The following procedure assists you in creating a new public distribution list ("group") or making changes to an existing group, stored on the Scalix server.

1 Start ADUC.

2 Locate a group record you want to review.

3 Open that record's [group name] **Properties** dialog box.



4 Click the **Scalix Server** tab (shown above).

5 Enter or modify the following required settings:

- Check **Enable Scalix Mailbox Services...** (if no checkmark is present).
- Open the **Home Node** menu and choose the mail node (if options are available) that this group record is associated with—if changes have been made.
- If the address in the **Default E-mail Address** field needs alterations, make them at this time.

6 Enter or modify the following optional setting:

- Click **Add E-mail** if there are other e-mail addresses associated with this group. A dialog box appears, in which you can type the address. (You can sort the addresses in this list, if more than one is listed.)

Note	The Default E-mail fields in both the General tab and Scalix Server tab are linked. Change one entry and the other will automatically update-- if you click **Apply**. If you change one, then open the other and change that without clicking Apply, problems may occur.

7 Click **Apply**, then click **OK** to save your entries and close the [group name] Properties dialog box.

Alert	DO NOT CLEAR the “Enable” checkmark unless you are sure you want to delete the mailbox along with all group-related data. That record will nominally remain in Active Directory, but there will be no data or mail.

After the next omldapsync operation, the changes will be in effect.

Deleting a Scalix mailbox in Active Directory

You can delete a user's or group's Scalix mailbox and all its contents, while retaining the user or group record in Active Directory. The following procedure will result in an Active Directory account record that used to be associated with a Scalix server mailbox; on completion, there will be no mailbox for the user or group, and all Scalix data will be deleted.

For example, you may want to perform this task after migrating the Scalix mailbox to another, separate server.

1 Start ADUC.

2 Locate a user or group record with the Scalix mailbox you want to delete.

3 Open that record's [name] **Properties** dialog box.

4 Click the **Scalix Server** tab.

5 Clear the checkmark in the **Enable Scalix Mailbox Services...** (if a checkmark is present).

Alert	DO NOT CLEAR the "Enable" checkmark unless you are sure you want to delete the mailbox along with all account-related data. That account record will nominally remain in Active Directory, but there will be no Scalix-based data or mail associated with it.

6 Click **Apply**, then click **OK**.

A confirmation dialog box appears.

7 Click **OK** to confirm and complete the mailbox deletion.

After the next omldapsync operation, the mailbox and its contents will be deleted. The Active Directory account will remain, for other uses.

Reviewing AD-managed Users and Groups with the Scalix Administrative Console

When you log into your Scalix server with the Scalix Administrative Console, you can review all of your Active Directory-managed users and groups, but the data fields for users or groups will be grayed out/inactive. You can see the entries and settings, but you cannot modify that data with SAC. You must use Active Directory to manage each of your Scalix users and groups.

Setting up Kerberos Authentication for the Scalix System

Kerberos, a network authentication protocol, provides strong authentication for client/server applications by means of secret-key cryptography. The Kerberos protocol relies on authentication tickets to validate users and/or services.

A Kerberos client (such as Scalix Connect for Outlook) can perform secure communications with a Kerberos service if both the client and the service authenticate against a *Key Distribution Center* (KDC) which is on the Kerberos Server. At this point, the client and service both obtain a *Ticket Granting Ticket* (TGT). The client then requests a service ticket for a specific service.

Therefore, a triangular relationship exists between the client and the KDC, the service and the KDC and between the client and the service. A Kerberos principal is either a client identity or a service identity operating in the Kerberos realm.

When an authorized user logs into your domain, a request is made by the e-mail client for a Ticket, and once authenticated, that user can use the ticket for as long as it remains valid. Server-side tickets are stored in the **keytab** file. Client-side tickets are stored in a temporary file.

In the Scalix environment, you can establish secure Kerberos communications for the following Scalix Services:

- Remote Execution Service
- Scalix Administration Console Service
- Scalix UAL Service
- Scalix IMAP Service

You can then configure Single Sign-on authentication with a KDC on the master domain controller that uses Microsoft Active Directory. This allows Scalix users to automatically authentication with the Scalix Server when they log in to their Windows domain.

To implement a Kerberos-based SSO process, you must have the following:

- Active Directory
- Scalix Connect for Microsoft Outlook
- Scalix Server
- the ktpass utility (available from the Microsoft Developer Network support web site

Installing ktpass

The ktpass utility creates Kerberos keytab files that can be used by Linux Kerberos-based systems to define Key Distribution Center (KDC) hosts and user/service mappings.

ktpass is available from the Windows 2000 resource kit, in this directory:

```
\Support\Tools\2000RKST
```

or from the Windows Server 2003 installation CD, in this directory:

```
\Support\Tools\Support.cab.
```

For more information on this tool, review the information at this URL:

```
http://www.microsoft.com/windows2000/techinfo/planning/security/
kerbsteps.asp
```

You must install the ktpass utility on the Microsoft Windows domain controller server.

Listing Scalix Servers in the DNS Server

The following procedure provides all installations of Scalix Connect for Outlook with a starting point for searching a user's mailbox. The results of the following will be the creation of C name aliases to your Scalix servers and the proper listing of Scalix servers in the DNS server.

1. Log into the Domain Controller computer as a Windows administrator with domain admin rights.
2. Choose Start > Programs > Administrative tools > DNS.
3. When the DNS window appears, open the **Forward Lookup Zones** list and create the needed zones for your domains.
4. Be sure to create host records for all Scalix Servers in the appropriate Forward Lookup Zone.
5. Open the **Reverse Lookup Zones** list and create the needed zones for your domain subnets.

   Once the zone/host records are ready, follow these steps:
6. Having already started DNS, reopen the **Forward Lookup Zones** list.
7. Right-click a Scalix Server Single Sign-on domain and choose **New Alias** from the pop-up action menu.
8. In the New Resource Record dialog box, in the **Alias** tab, make these entries:
   - Enter "scalix-default-mail" (without quotes) in the **Alias name** field.
   - Enter the fully-qualified name of the Scalix Server in the **Fully qualified name for target host** field. (for example, scalixserver.acme.net)
9. Click **OK**.
10. Open the Reverse Lookup Zones list.
11. Right-click the subnet in which the Single Sign-on Scalix Server resides, and choose **New Pointer** from the pop-up action menu.
12. In the New Resource Record dialog box, in the **Pointer** tab, make these entries:
    - Enter the last two or three digits of the Scalix server IP address.
    - Enter the fully-qualified hostname of the Scalix Server.
13. Click **OK**.
14. Close the DNS window.

Using the ADUC Utility

1 Choose Start > Programs > Administrative Tools > **Active Directory Users and Computers**.

2 Review the list for any "Scalix services" records/entries/items.

3 If one does not already exist, right-click the root domain controller and choose New > **Organizational Unit**.

4 When the New Object dialog box appears, click in the **Name** field and type "Scalix Services" (without the quotes).

5 Click the now-active **OK** button.

This creates a separate organizational unit (OU) for Scalix server data.

6 In the UC window, right-click this new Scalix Services organizational unit and choose New > **User** from the pop-up action menu.

7 When the New Object-User dialog box appears, make these entries:

- Click in the **First Name** field and type "scalix-ual" (without the quotes).
- Click in the **Last Name** field and type the name of the Single Sign-on Scalix Server. This allows you to identify the keytabs you generate for multiple Scalix Servers.
- Do not modify the default selection in the **User logon name** pull-down menu.
- Click in the **User logon name** field and type "scalix-ual-[name]" (without the quotes).

Replace the "[name]" placeholder with a unique ID for this server, so that you can create other scalix-ual-[name] users for other Scalix servers that you want to include.

8 Click **Next**.

9 When the Password features appear in the New Object-User dialog box, enter and confirm a password for the user. Make sure that the password you enter is sufficiently complex

- Leave the **User must change password at next logon** option unchecked.
- Leave the **User cannot change password** option unchecked.
- Check the **Password never expires field** checkbox.

10 Click **Next**.

11 Click **Finish**.

12 As a result, an Active Directory "user" now represents the Scalix UAL Service for the Scalix server.

Converting a Scalix Service account to Kerberos Service

To change the Scalix Service account to a Kerberos Service account and generate a keytab, follow these steps:

1 After logging into the AD server, open a Command Prompt window.

2 Navigate to the directory that contains ktpass.

In most cases, you'll find it in c:\Program Files\Support Tools.

3 If ktpass does not automatically load, enter this command at the prompt:

```
ktpass –princ scalix-ual/scalixservername.domain@REALM –mapuser
scalix-ual-[name] -pass password –out path\filename -kvno 3
```

For example:

```
ktpass -princ scalix-ual/scalixserver.acme.net@ACME.NET -mapuser
scalix-ual-[name] -pass password -out scalix-ual-[name].keytab -
kvno 3
```

IMPORTANT: The REALM entry MUST be uppercase.

Note where "scalix-ual" and "scalix-ual-[name]" are used in this syntax.

Note The -kvno option prevents potential key version mismatches that can cause SSO to fail. Setting this value to 3 ensures that the keytab version is the same for both existing and future users in Active Directory.

The following status messages appear, to indicate that the keytab was successfully created:

```
Successfully mapped scalix-ual/scalixserver.acme.net to scalix-
ual.

Key created.

Output keytab to scalix-ual.keytab:

Keytab version: 0x502

keysize 68 scalix-ual/scalixserver.acme.net@ACME.NET ptype 1
(KRB5_NT_PRINCIPAL)

vno 3 etype 0x1 (DES-CBC-CRC) keylength 8 (0xe6fb762ad01f8a9b)

Account has been set for DES-only encryption.
```

4 Copy this keytab (file) to the home directory of the Single Sign-on Scalix Server.

5 On the Scalix Server, log in as a root administrator.

6 To merge the keytab you created with the Kerberos system keytab file, type this command:

```
ommergekeys /path/filename.keytab
```

7 Modify the /etc/krb5.conf file by means of this command:

```
omkrbconf -r REALM -s servername.domain -d domain
```

IMPORTANT: The REALM information MUST be uppercase.

Use the following arguments/extensions for this command:

- **-r** specifies the realm that the Kerberos database controls. For example, if your domain name is acme.com, your realm is ACME.COM.

- **-s** specifies the fully qualified host name of the Kerberos KDC machine. For Single Sign-on, the KDC is the Domain Controller with Active Directory installed.
- (optional) **-d** specifies the domain name in which the Kerberos Realm operates. If you do not specify a value, the domain name is determined from the current domain.

8 Exit the Command Prompt window.

This completes the Kerberos secure communication setup.

Matching authentication IDs and domain identities

In order for Single Sign-on to fully function, the authentication ID for a Scalix Server mailbox must match the domain identity (the ID in Active Directory) for the user. For example, if jsmith@acme.net is the User Logon ID for a user in Active Directory, enter the following on the Scalix Server:

```
ommodu –o jsmith ––authid jsmith@ACME.NET
```

To view the authid value (-o) for a user, enter this command:

```
omshowu "Joe Smith/mailnode"
```

This user can now use Single Sign-authentication. After the user logs into the Windows domain, the user no longer must enter username or password information during MS Outlook profile creation or login.

If Active Directory is unavailable at any point after setting up Single Sign-on, the Scalix Server prompts users for their regular Scalix-domain password for authentication.

A Complete List of the Scalix Extensions of Active Directory

Scalix has been given the OID root of 1.3.6.1.4.1.19049, and all of the following extensions are appended to it. The first table shows the extensions that match the options in the Scalix Server tab (esp. users)

[1.1.10] scalixScalixObject	True if this is an object managed by Scalix.
[1.1.11] scalixMailnode	The mail node that is hosting this object.
[1.1.12] scalixAdministrator	True if this user has general admin capabilities.
[1.1.13] scalixMailboxAdministrator	True if this user has mailbox admin capabilities.
[1.1.14] scalixServerLanguage	The language for server to client communications.
[1.1.15] scalixEmailAddress	A multivalued list of email addresses for this mailbox.
[1.1.16] scalixLimitMailboxSize	The maximum size of the mailbox in MB -- 0 to use server default.
[1.1.17] scalixLimitOutboundMail	True if Scalix will warn when near limit on outbound mail.
[1.1.18] scalixLimitInboundMail	TRUE if Scalix will reject inbound mail upon limit reached.
[1.1.19] scalixLimitNotifyUser	TRUE if Scalix will notify user when limit is reached.
[1.1.20] scalixHideUserEntry	TRUE if this directory entry is to be hidden from the CDA.
[1.1.21] scalixMailboxClass	Set to “full” or “limited” to control class, or leave it blank for default.

The following table shows the Scalix object classes that extend the Active Directory OpenLDAP schema..

[1.2.10.23] scalixUserClass	Auxiliary class of attributes to extend the User class
[1.2.11.24] scalixGroupClass	Auxiliary class of attributes to extend the Group class

Chapter 9

Directory Synchronization

Directory Synchronization is the process of automatically exchanging entries between Directories. Directory Synchronization agreements manage how the entries are exchanged, and the process includes the use of the Directory Synchronization Server (DS Server).

This process ensures that when you add, delete, or modify an entry in one Directory, the change is applied to all other Directories in your messaging network.

Note	Support for multiple Scalix servers and directory synchronization is available only in Scalix Enterprise Edition. For more information, see "About Scalix Product Editions".

This chapter includes the following information:

Overview

Directory Synchronization allows you to automatically maintain consistent Directory entries across a network. It ensures that whenever you add, modify, or delete an entry at its primary location, the change is applied to other Directories throughout the network. Directory Synchronization ensures the following:

- Directory entries are always up-to-date throughout the network.
- Fewer messages are incorrectly addressed.
- A minimal amount of time is required to maintain Directories.

You can synchronize Directories with other Scalix Directories or with Directories in other mail systems.

Note	This chapter only describes Scalix-to-Scalix Directory synchronization.

Note that synchronization is not possible with Directories acting as X.500 Directory access points (marked as X500 in the omlistdirs command).

Synchronization occurs by sending mail messages between systems. The Directory Synchronization Server (DS Server) on each system generates and processes these Note that synchronization is not possible with Directories acting as X.500 Directory access points (marked as X500 in the omlistdirs command).

Synchronization Messages

A single system can contain more than one Directory, and Directory Synchronization is established between Directories (not necessarily between systems). Directory Synchronization is controlled through one-to-one Directory Synchronization agreements between Directories. In each case, one Directory is the export Directory and the other is the import Directory.

In a synchronization agreement between two Scalix Directories, the importing DS Server requests Directory updates from the exporting DS Server. The exporting DS Server extracts Directory updates from the relevant Directory change log and returns the updates to the importing DS Server where they are applied to the import Directory. This process is automatically repeated at set intervals, with the importing DS Server always initiating the exchange of messages.

Often, two synchronization agreements are created between a pair of Directories so that a bidirectional link is created. In this scenario. each Directory acts as both an import and an export Directory.

Scalix Corporation recommends that the Scalix Network Administrator research and design the optimal network topology to use for Directory Synchronization.

For each Directory Synchronization agreement, the import and export Directory must be consistent in terms of the following:

- The names of the Directories to be synchronized
- The addresses of the importing and exporting DS Servers
- The frequency of the updates

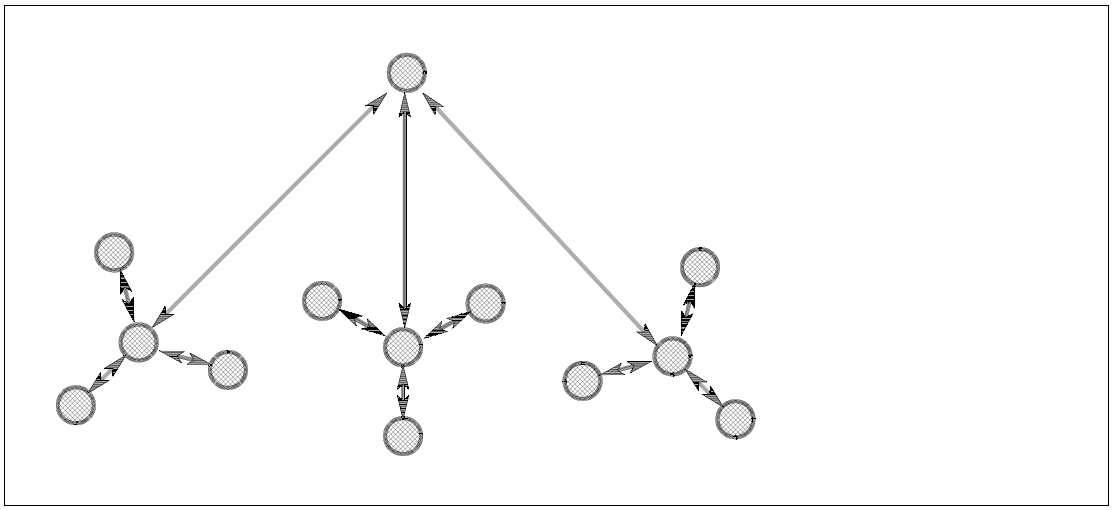
For every local import agreement there must be an associated agreement on the exporting system.

Also, you might have to update the Routing Table on each system to provide routes to the new O/R Addresses that are made available through Directory Synchronization.

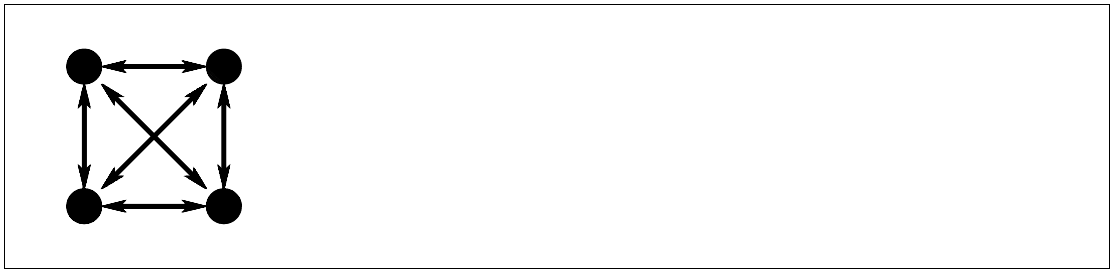
Synchronization Topologies

Directory Synchronization can operate with any network topology. Directory Synchronization is flexible and allows you to plan the overall topology in a manner that best suits your organization. Directory Synchronization topology does not need not correspond to the physical network topology of the systems involved.

In many cases, a hub network for Directories is the easiest topology to create and maintain. The following illustration shows a hub network.



However, in a small network with only a few Directories, you might want to create one-to-one Directory synchronization agreements between every pair of Directories as shown in the following illustration.



Scalix-to-Scalix Directory Synchronization

Directory Synchronization is the process of automatically exchanging entries between Scalix Directories in a Directory Synchronization agreement. The process is performed using the Directory Synchronization Server (DS Server).

Directory Synchronization ensures that whenever you add, modify, or delete an entry in one Directory, the change is applied to other Directories throughout the network

The synchronization process is configured and controlled using the commands omaddds, omdelds, ommodds, omresyncds, omlistds, and omshowds. See “Directory Synchronization Commands” on page 138 for information about these commands.

Import and Export Sequence

Periodically, the local DS Server (the importing server) sends a mail message to another DS Server (the exporting server) requesting updated entries in the exporting Directory. The exporting DS Server replies with one or more messages containing the updated entries. The importing DS Server then applies these entries to the importing Directory.

The entries to be exported are identified by a time stamp in the Directory change log of the exporting Directory. All entries with a time stamp later than the time of the previous import cycle are exported.

If the time of the previous import cycle is earlier than the first entry in the change log, all entries in the export Directory are exported. All entries in the export Directory are exported when

- synchronization is first started
- when the omresyncds command is issued
- and when the period between successive successful import cycles is longer than the configured change log life.

See “Synchronization Rules” on page 126 for a description of the different messages and the sequence in which the messages are sent.

Masks Filters and Selectors

When exporting Directory entries, you can configure the DS Server to mask (remove) specified Directory attributes from an entry. You can use this feature to prevent sensitive information from being exported.

When importing Directory entries, you can filter entire entries and prevent an imported entry from modifying or deleting an existing entry.

When both exporting and importing entries, you can also explicitly select the entries you want to export or import.

Masks, filters, and selectors are defined using the omaddds or ommodds commands.

See “Directory Synchronization Commands” on page 138 for information about these commands. You can also view information about Scalix commands in the online manual page.

Synchronization Rules

When importing Directory entries from another Scalix Directory, the following rules are applied in addition to any configured selectors and filters:

- An entry is added to the Directory if the primary attributes of the entry do not match an existing entry.

An existing entry is modified or deleted if:

- its primary attributes match those of an imported modifying or deleting entry
- the existing entry was previously imported from the same Directory as the imported modifying or deleting entry (the PROPAGATED-BY attributes match).
- If the -I flag is set by the omaddds or ommodds commands, an existing entry is modified or deleted if its primary attributes match those of any imported modifying or deleting entry.

When exporting Directory entries to another Scalix Directory, the following rules are applied in addition to any configured masks and selectors:

- An entry is not exported if it was previously imported from the Directory to which it is being exported (the PROPAGATED-BY attribute matches the LOCAL-UNIQUE-ID attribute of the importing agreement).

- An imported entry that is manually modified or deleted (a modification or deletion not performed by the local DS Server) is not exported unless the -E flag is set by the omaddds or ommodds commands.

The PROPAGATED-BY Attribute

The rules applied when importing and exporting Directory entries use the PROPAGATED-BY attribute (internal tag 321). This attribute identifies the Directory from which the entry was imported. If there is no PROPAGATED-BY attribute, the DS Server interprets the entry as being owned by the local system.

The PROPAGATED-BY attribute is added to an entry when it is imported. Its value is that of the LOCAL-UNIQUE-ID attribute (internal tag 308) of the Directory entry defining the import agreement.

Use the omsearch command with the -m @ALL-ATTR@ option to display the LOCAL-UNIQUE-ID of an entry. For example, to display all Directory entries in the Directory PINEWOOD that define import agreements, enter:

```
omsearch -d PINEWOOD -e "DS-SYNC-DATA=1" -m @ALL-ATTR@
```

Other values for DS-SYNC-DATA are:

- 2 for export agreements
- 3 for Directory-wide configuration data

The PROPAGATED-BY attribute relates to the import agreement on the local system. The attribute is not exported and does not identify the system on which the entry originated. The PROPAGATED-BY attribute only identifies the system from which it was imported through the information contained in the import agreement.

Using the DS_SEND_SOURCE_LID Option

The DS_SEND_SOURCE_LID option in the general configuration file enables a unique identifier to be propagated with each entry. The value of this identifier is that of the LOCAL-UNIQUE-ID of the entry in the exporting Directory.

For more information, see “Configuration Options” on page 301.

Adding Attributes to Entries

When you add an entry to a Directory, the Country Name, ADMD, PRMD, and Organization Name attributes in the O/R Address configured for the exporting DS Server are added to the entry if they are not already present. Only attributes of a higher significance than those that already exist are added.

The addition of these attributes ensures that if the entry was imported through an X.400 Interface, the O/R Address in the imported entry can be routed correctly.

For example, if an entry containing the O/R Address;

```
Marion Brand/london,sales,mis
```

is imported through the local X.400 Interface, and the O/R Address of the exporting DS Server is;

```
+DIRSYNC/london,sales,mis/pinewood/gb/gold 400/forester
```

then the O/R Address in the imported entry for Marion Brand is;

```
Marion Brand/london,sales,mis/pinewood/gb/gold 400/forester.
```

Using the omresyncds Command

The omresyncds command causes entries in the specified local (or importing) Directory to be resynchronized with entries in export Directories with which it has an import Directory Synchronization agreement. When importing entries, the omresyncds command maintains the cross-references between PDL members and Directory entries.

The omresyncds command performs the following actions in the importing Directory:

- Updates entries where the value is different from the corresponding value in the exporting Directories.
- Updates PDL members cross-referenced to updated entries.
- Deletes any Directory entries (including PDL members) that no longer have a corresponding entry in an exporting Directory.

Synchronization Agreement Guidelines

Note the following guidelines when using synchronization agreements between Scalix Directories:

- To disable a specific import agreement, rather than the whole synchronization process, set the "start at time" to zero (ommodds -i number -t 0), and then stop and restart the DS Server.

To disable a specific export agreement, disable its corresponding import agreement. If this is not possible, delete the export agreement, and then stop and restart the DS Server.

- If you used a manual process to synchronize an existing Directory, do not try to synchronize the Directory using the DS Server. The ownership information in the entries will be incorrect.

Instead, create a new Directory containing only those entries that you want to control (the entries must not contain the PROPAGATED-BY attribute). Then, allow the synchronization process to add the other entries.

- If you make a mistake while setting up a synchronization process that results in incorrect entries being added, deleted, or modified from a Directory, begin the synchronization process again using a new Directory containing new entries. The new entries must only be those that you want to control and must not contain the PROPAGATED-BY attribute. After you do this task, allow the synchronization process to add the other entries.
- When setting up an agreement, consider your network topology in relation to the timing of your imports and exports. For example, in a hub topology, you might want to export/import entries to/from the hub system during low-usage periods. For more information, see the section “Synchronization Topologies” on page 124.

The timing of imports and exports is managed by the synchronization agreements using the "start download time", "start upload time", and "period" options.

- When testing a synchronization process, use the following general.cfg options to accelerate the synchronization cycle:

DS_CUST_MSGQ_TIMEOUT

DS_CUST_SEND_REQ_NOW

DS_CUST_PERIOD_TIMER_MINUTES

For more information, see “Configuration Options” on page 301.

- If you change the primary mailnode on a system, you must modify all agreements on all systems referencing the mailnode. For more information, see “Primary Mailnode” on page 6.

Message Types and Sequences

Messages sent between DS Servers are mail messages containing one IA5 body part with line lengths of 80 characters or less. Directory entries containing non-IA5 characters are represented by nnn, where nnn is the decimal-coded value of the character. Directory entries, attribute selectors, and filters of more than 80 characters are split across lines in a message and reassembled by the recipient DS Server.

The following messages can be sent between DS Servers during Scalix to Scalix Directory Synchronization:

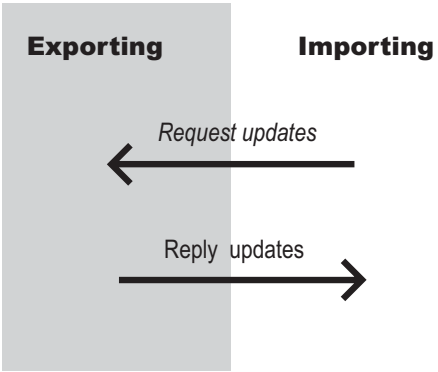
- Request Updates/All
- Reply Updates/Reload/All
- Indication
- Fault Updates/All

In addition, the following messages can be sent from an importing DS Server to the local Error Manager:

- Error Reply Updates/All
- Error Fault Updates/All

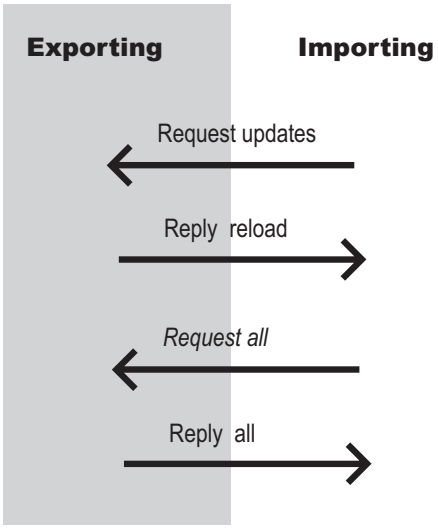
Request Updates Message

An importing DS Server periodically sends a Request Updates message to an exporting DS Server. The exporting DS Server returns a Reply Updates message containing all entries in the relevant Directory change log that have a time stamp later than that supplied in the Request message sent by the importing DS Server. The following illustrations shows the request message update flow



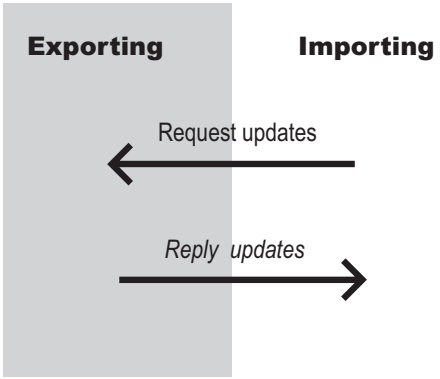
Request All Message

An importing DS Server sends a Request All message to an exporting DS Server in response to a Reply Reload message. The exporting DS Server returns a Reply All message containing all entries in the relevant Directory. The following illustration shows the request all message flow.



Reply Updates Message

An exporting DS Server sends a Reply Updates message to an importing DS Server in response to a Request Updates message. The Reply Updates message contains all entries in the relevant Directory change log that have a time stamp later than that supplied in the Request message. The following illustration shows the reply updates message flow.



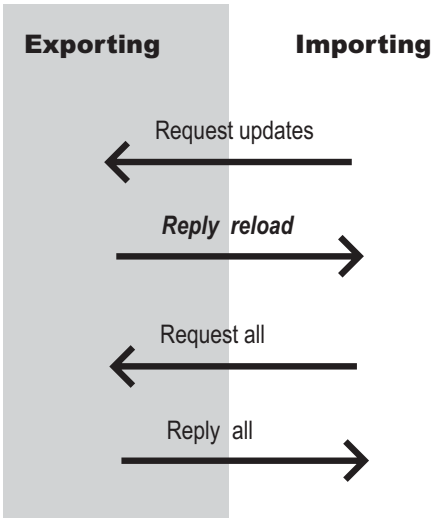
Reply Reload Message

An exporting DS Server sends a Reply Reload message to an importing DS Server in response to a Request Updates message when the time stamp specified in the Request message predates the first entry in the relevant Directory change log.

The Reply Reload message notifies the importing DS Server that it is required to reload the relevant Directory. The importing DS Server responds with a Request All message. A reload is required because:

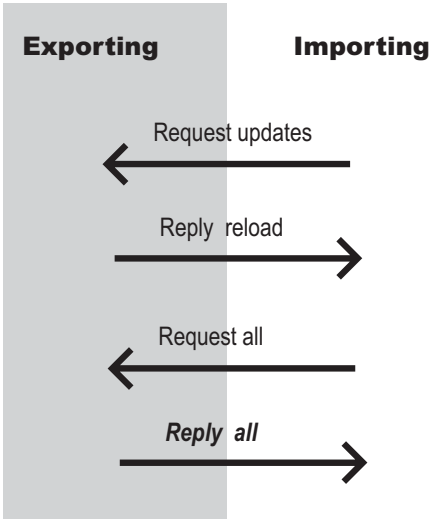
- Entries added to the Directory change log since the last Reply Updates message might have been overwritten
- The omresyncds command was executed

The following illustration shows the reply reload message flow.



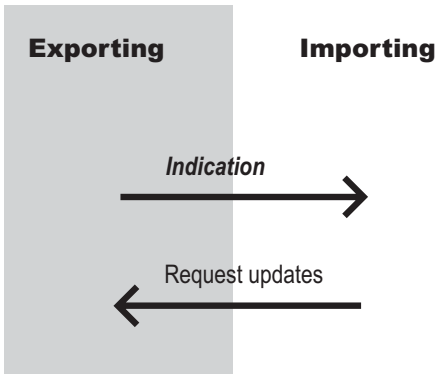
Reply All Message

An exporting DS Server sends a Reply All message to an importing DS Server in response to a Request All message. The Reply All message contains all the entries in the relevant Directory. The following illustration shows the request all message flow. The following illustration shows the reply all message flow.



Indication Message

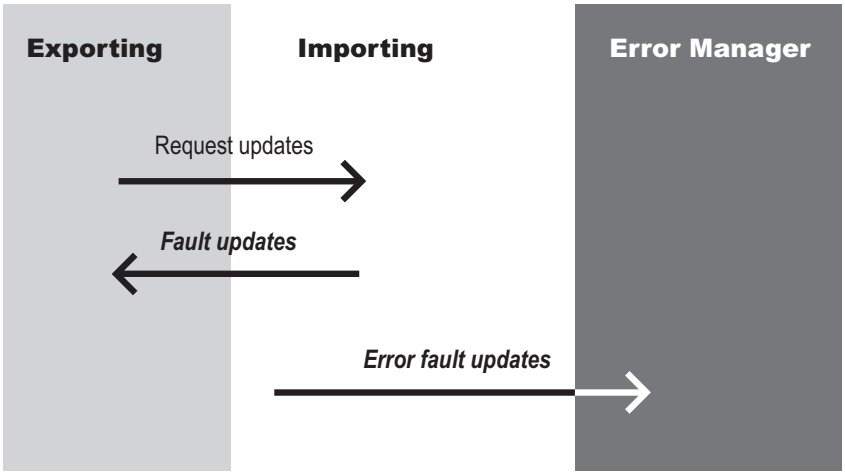
An exporting DS Server sends an Indication message to an importing DS Server when the importing DS Server does not send a Request Updates message for a user-specified period of time. The Indication message notifies the importing DS Server to send a Request Updates message. An Indication message is sent only if there are new entries in the relevant Directory change log. The following illustration shows the indication message flow.



Fault Updates Message

An exporting DS Server sends a Fault Updates message to an importing DS Server in response to a Request Updates message when the exporting DS Server cannot process the Request.

The Fault Updates message sends the cause of the Request processing failure to the importing DS Server. The importing DS Server then sends an Error Fault Updates message to the Error Manager for the importing DS Server system. The following illustration shows the fault updates message flow.

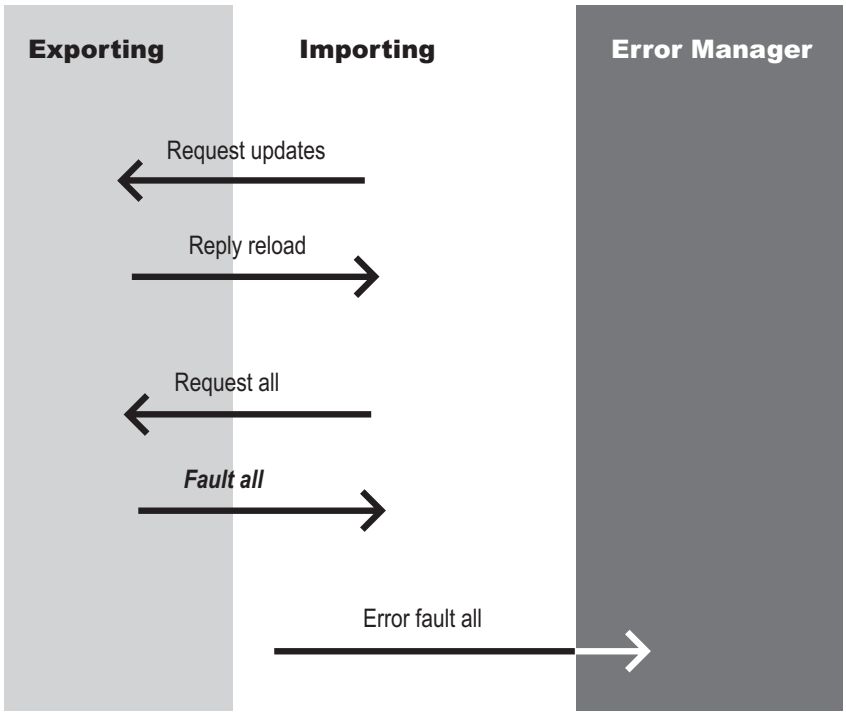


Fault All Message

An exporting DS Server sends a Fault All message to an importing DS Server in response to a Request All message when the exporting DS Server cannot process the Request.

The Fault All message sends the cause of the Request processing failure to the importing DS Server. The importing DS Server then sends an Error Fault All message to the Error Manager

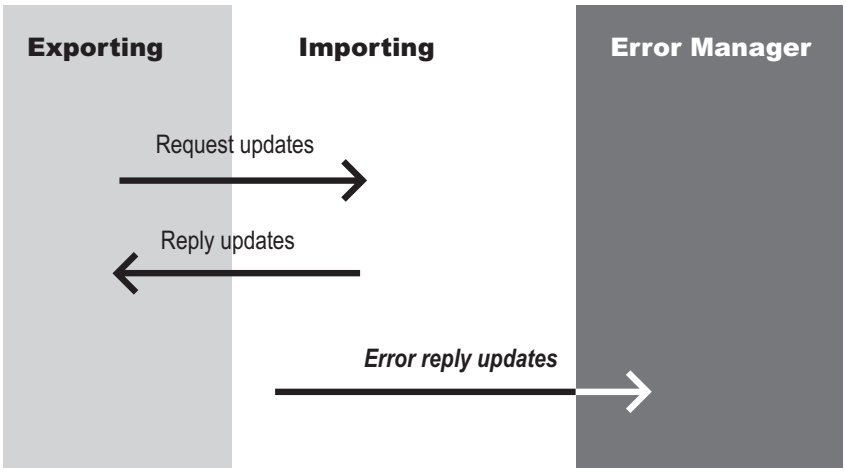
for the importing DS Server system. The following illustration shows the fault all message flow.



Error Reply Updates Message

An importing DS Server sends a Error Reply Updates message to the Error Manager for the importing DS Server system when a Reply Updates message cannot be processed by the importing DS Server.

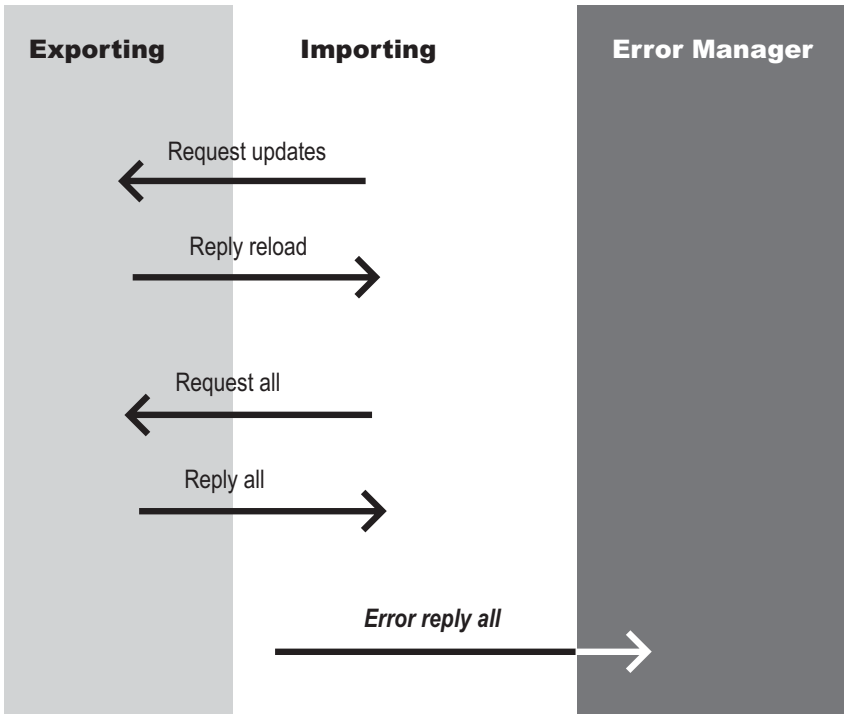
A non-fatal Error Reply Updates message contains the Directory entries that the importing DS Server cannot process. A fatal Error Reply Updates message contains the first entry that cannot be applied. The following illustration shows the error reply updates message flow.



Error Reply All Message

An importing DS Server sends a Error Reply All message to the Error Manager for the importing DS Server system when a Reply All message cannot be processed by the importing DS Server.

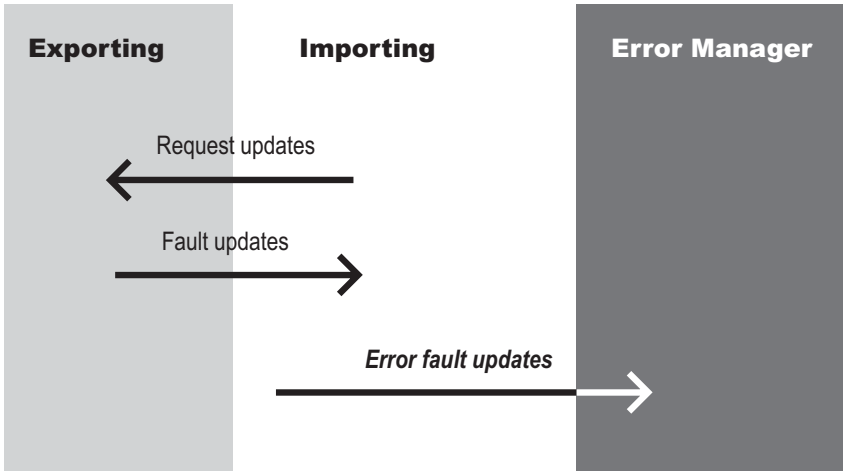
A non-fatal Error Reply All message contains the Directory entries that the importing DS Server cannot process. A fatal Error Reply Updates message contains the first entry that cannot be applied. The following illustration shows the error reply all message flow.



Error Fault Updates Message

An importing DS Server sends an Error Fault Updates message to the Error Manager for the importing DS Server system when a Fault Updates message is received from the exporting DS Server. An Error Fault Updates message contains the cause of the processing failure for the

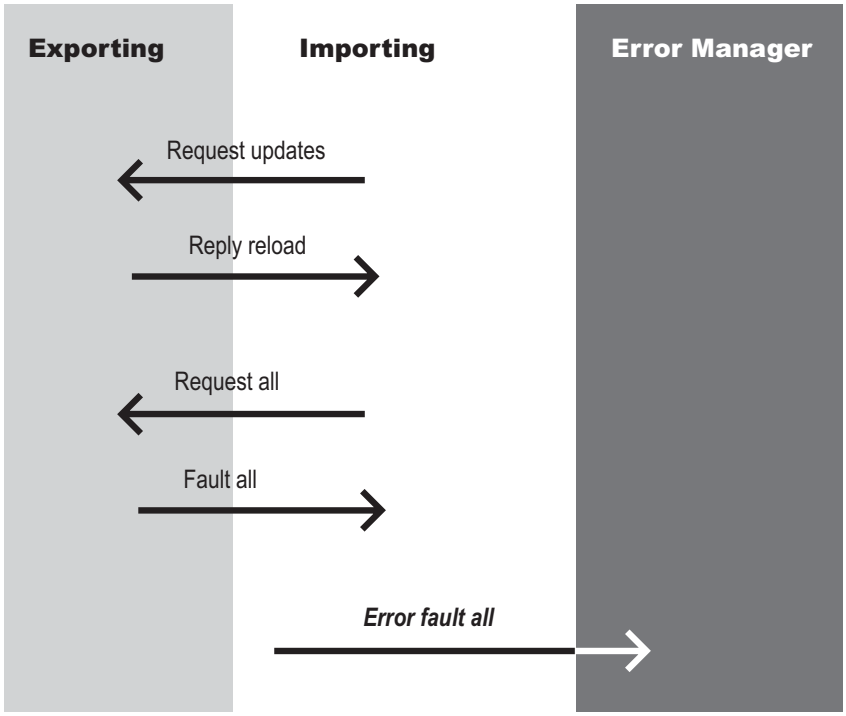
original Request message on the exporting DS Server. The following illustration shows the error fault updates message flow.



Error Fault All Message

An importing DS Server sends an Error Fault All message to the Error Manager for the importing DS Server system when a Fault All message is received from the exporting DS Server.

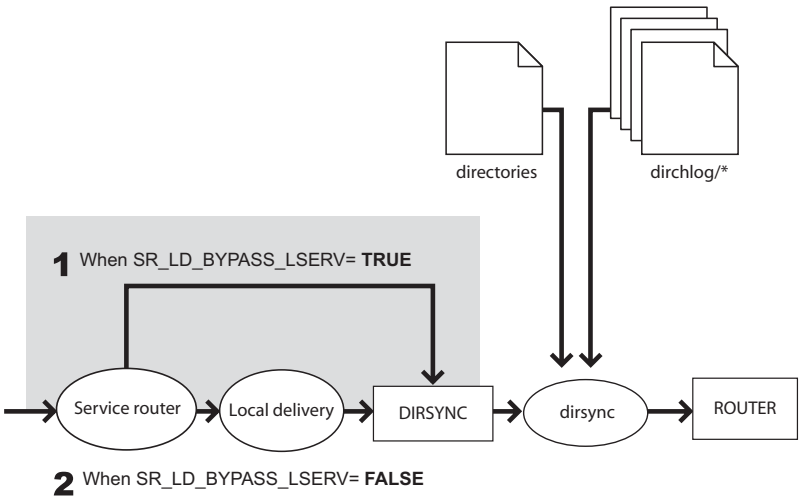
An Error Fault All message contains the cause of the processing failure for the original Request message on the exporting DS Server. The following illustration shows the error fault all message flow. The following illustration shows the error fault all message flow.



Directory Synchronization Server

The Directory Synchronization Server (DS Server) implements the protocols used to synchronize entries between Directories, and exists on all primary mailnodes in a Scalix network. The Directories can be on the same system or on different systems.

A DS Server sends Directory entries and requests for Directory entries to one or more DS Servers using standard e-mail messages. The DS Server automatically generates and processes the messages. A DS Server can both import and export Directory entries.



The Directory Synchronization Server routing process is as follows:

1 Messages addressed to a non-local DS Server are routed by the Service Router.

If a message is local and addressed to +DIRSYNC, Scalix routes the message directly to the DS Server queue DIRSYNC if the SR_LD_BYPASS_LSERV option is set to TRUE (default). See “Configuration Options” on page 301 for more information.

If the SR_LD_BYPASS_LSERV option is set to FALSE, the Service Router routes the message to the Local Delivery Service. The Local Delivery Service then routes the message to the DS Server queue. See “Configuration Options” on page 301 for more information.

2 The DS Server receives the messages from the DIRSYNC queue and performs the operation specified in the last body part of the message.

3 The DS Server attaches messages that it generates to the ROUTER queue of the Service Router.

The following table lists key Directory Synchronization Server information.

Item	Name
Service	dirsync
Process	dirsync
Queue	DIRSYNC

Item	Name
Surname	+DIRSYNC

Directory Synchronization and Security

In addition to password control on Directories, access to the Directories and the DS Server is controlled by Access Control Lists (ACLs). The DS Server must have update permissions specified in the ACL in order to update a Directory entry. The DS Server accesses Directories as a local Administrator and has update permissions enabled by default.

The DS Server must also give use permissions to other DS Servers from which it is receiving messages. All users have use permissions to the DS Server by default. Also, messages containing Directory entries are only accepted by an importing DS Server if the messages are in response to requests from that server.

Directory Change Logs

Each export Directory in a Directory Synchronization agreement has an associated Directory Change log. When an importing DS Server requests an update, the exporting DS Server uses the information in the Change log to determine the entries that must be sent to the importing DS Server.

When a Directory entry is added, deleted, or modified, the change is recorded in the Change log for that Directory. Each entry in the change log consists of:

- the Directory entry itself, the type of update (add, delete, modify)
- the source of the update (Directory Synchronization or other method)
- the source of the original entry (Directory Synchronization or other method)
- the time (in seconds since 1/1/1970) that the update occurred

A change log is created when the first change is made to a Directory after an export agreement is created for that Directory. Change logs are in the ~/dirchlog directory. Each change log consists of two text files and a lock file. New entries are written to the second text file. When the second text file expires (as specified by the omaddds or omaddss command), the contents of the second file are copied to the first text file.

The change log for a specific Directory can be identified by viewing the file ~/sys/dir.index. The third field in this file displays the name of the Directory and the fifth field gives the prefix of the change log file names. The age of the change log is shown in the last field (in seconds).

You can determine the amount of data written to a change log for changes made to PDL members by setting the DR_NO_MOD_STRIP_PDL option in the General Configuration File (~/sys/general.cfg). See “Configuration Options” on page 301 for more information.

Directory Synchronization Commands

The following table lists and describes commands associated with Scalix Directory Synchronization.

Command	Description
omaddds	Add a Directory Synchronization agreement.
omdelds	Delete a Directory Synchronization agreement.
omlistds	List Directory Synchronization agreements.
ommodds	Modify a Directory Synchronization agreement.
omresyncds	Resynchronize a Directory.
omshowds	Show details of a Directory Synchronization agreement.

Chapter 10

Public Distribution Lists

This chapter describes PDLs, which enable you to specify just one address to send messages to groups of users. This chapter includes the following information:

PDL Overview

A Public Distribution List (PDL) is a Scalix Directory entry. It is identified and accessed by using an O/R Address, and contains a list of O/R Addresses.

PDL Directory entries usually contain the O/R Addresses of local and remote users, but they also can contain the O/R Addresses of other PDLs. PDLs enable you to specify just one address to send messages to groups of users.

Access Control Information (ACI) can be associated with a PDL to control which users can access the PDL or change its contents.

When a message addressed to a PDL reaches the Scalix system holding the PDL, the PDL is expanded (that is, the PDL address is replaced by the list of O/R Addresses in the PDL).

A PDL is expanded during message delivery only if the PDL is held in the system default Directory. Scalix cannot expand PDLs held in Directories other than the system default Directory.

PDLs are expanded during message delivery usually by the Local Delivery Service. However, PDLs can also be expanded by the Service Router. If the Service Router is unable to route the O/R Address of the PDL, it will resolve the address, and in doing so, expand it.

The commands omaddpdl, omdelpdl, ommodpdl, and omshowpdl add, delete, modify, and list PDLs on the local system, respectively. The commands omaddpdln, omdelpdln, and omshowpdln add, delete, and list O/R Addresses in a PDL on the local system.

The omaddpdln command adds O/R addresses to a PDL. Each time omaddpdln is used to add an address to a PDL, only one record is added to the Directory change log.

The command ommodpdln modifies O/R addresses in a PDL. If you need to modify PDL entries which are Cross-referenced (see “Cross-referenced Entries in PDLs” on page 140), you do not need to use this command. Instead, use the ommodent command to directly modify the Primary entry.

An example Linux script to generate a PDL for all users on the local system is under the directory /opt/scalix/examples/general.

PDL Directory Entries

A Directory entry for a PDL contains O/R Address attributes identifying the PDL, including:

- its name
- a compound attribute DL-MEMBERS (internal tag 2.6.5.2.3) containing a list of O/R Names (usually O/R Addresses) representing the members of the PDL.

For example:

```
S=Lawyers/G=Company/OU1=ny/OU2=legal/
ENTRY-TYPE=4/LOCAL-UNIQUE-ID=1234/
DL-MEMBERS=[S=Armstrong/G=Jay/OU1=paris/OU2=legal/OU3=linux]=
[S=Baldon/G=Sam/OU1=london/OU2=legal]=
```

The entry in the Directory is identified as a PDL by the attribute ENTRY-TYPE (internal tag 304) being set to 4.

Cross-referenced Entries in PDLs

If an O/R Address listed within a PDL also exists as a primary entry in the same Directory, the address in the PDL is cross-referenced to its primary entry when the address is added to the PDL. When an address within a PDL is cross-referenced, any changes to its primary entry are automatically reflected in the address held within the PDL.

The cross-reference between the PDL entry and the primary entry is maintained using the PARENT-DL attribute (internal tag 423) in the primary entry. The PARENT-DL attribute holds the LOCAL-UNIQUE-ID value (an attribute that uniquely identifies an entry; internal tag 308) of any PDLs cross-referenced to it.

Using a Public Distribution List

PDLs have a wide range of different uses. They can hold the addresses of everyone in a department, the members of a committee, or all the employees in a company.

For example, if a PDL is created with the Personal Name “Company Lawyers” and local mailnode of OU1=ny/OU2=legal, then users in the network wanting to send a message to all the lawyers in the company could address their message to the following:

```
Company Lawyers/ny, legal
```

When the message arrives on the system with the local mailnode of OU1=ny/OU2=legal, the PDL in the message is expanded. The PDL address is replaced by the O/R Addresses listed in

the Directory entry for the PDL. The message is then sent on to all the addresses in this expanded list.

Adding a PDL to the Default Directory

You can add a Public Distribution List (PDL) to the default Directory using the command:

```
omaddpdl -l PDL_address [-a alias]...
```

Adding an Entry to a PDL in the Default Directory

You can add an entry to a Public Distribution List (PDL) in the default Directory using the command:

```
omaddpdln -l PDL_address [-n PDL_entry]
```

To add entries to a PDL in a Directory other than the system default Directory, use the omaddpdln -d command.

Access Control Information (ACI)

Access Control Information (ACI) is a compound attribute ENTRY-ACI (internal tag 510) in a PDL Directory entry, defining access control for the PDL. An ACI defines which users can access or delete a PDL Directory entry or change its members. An ACI can be thought of as an Access Control List (ACL) governing user access to PDLs. (For more information about ACLs, see “Access Control Lists” on page 197).

The commands omaddaci, omchkaci, omdelaci, ommodaci, and omshowaci enable you to add members to an ACI, check a user’s ACI permissions, delete members from an ACI, modify members of an ACI, and show the contents of an ACI, respectively. ACI commands look similar to ACL commands, but they are implemented differently. See “ACI Permissions for PDL and Directory Commands” on page 144. for details of these commands.

Note	If a PDL has an ACI associated with it, you must use ACI commands, not Directory commands, to manipulate the ACI. A user must have certain permissions defined in the ACI to use Directory commands.

When you create an ACI, you assign access permissions for the specified user or group of users. (The Linux root user and the Directory owner always have all permissions.) These permissions control who can configure an ACI as well as who can access the specified PDL entry and its members.

You assign these access permissions when you create an ACI with the command omaddaci -c. The following table lists the permissions you can assign.

Permission	Description
config	Enables or prevents a user configuring, or setting up, an ACI. All ACI commands (except omchkaci) require that the user has this permission.

Permission	Description
modify	Enables or prevents a user modifying the PDL Directory entry (that is, adding, deleting, or changing members of the PDL, or changing the O/R Address of the PDL). A user must also have `read` permission in order to be able to modify a PDL Directory entry.
read	Enables or prevents a user searching or reading the PDL Directory entry. You must assign this permission to a user to whom you are assigning any other permissions, because many of the operations using PDL commands implicitly include READ operations. For example, the `omaddpdln` command fails if a user has modify permission but no read permission because the `omaddpdln` must first read an entry before it can modify it.
remove	Enables or prevents a user removing a PDL Directory entry (not removing members of the PDL). A user must also have `read` permission to remove a PDL Directory entry.

These permissions apply both to the Scalix commands and to UAL client sessions.

For Scalix clients that use sorted lists of Directory entries provided by the Client Directory Access (CDA) Server, PDLs for which the user does not have the required permissions defined in the ACI might appear not to exist.

The CDA Server uses a snapshot of the real Directory to build the sorting information. A series of pointers is set up to the real Directory, and the snapshot is then discarded. The sorting information always includes all PDL Directory entries, regardless of defined permissions, because this information is shared by all clients. However, when a user who does not have the required ACI permissions tries to access a PDL, the CDA Server returns the substitute text <Deleted Entry>.

For more information about CDA, see “Client Directory Access” on page 98.

Routing PDLs Containing ACIs

When a message is addressed to a PDL, the client confirms that the PDL is valid and sends the message. When the message arrives at the Service Router or Local Delivery Service, these services access Scalix Directories to route and expand PDLs. For more information, see “How the Service Router handles messages” on page 69.

If an ACI is defined for the PDL, the services check the ACI to determine if the message originator has read permission to access the PDL:

- If the user does have read permission, the services continue to route the message.
- If the originator of the message does not have read permission defined in the ACI for the recipient PDL, a Non-Delivery Notification is generated with the error:

```
recipient name not found
```

The client interface determines if and how Non-Delivery Notification reports are presented to the user.

ACI Addresses and Groups

User addresses and user groups within an ACI work similarly to addresses and groups within an ACL. The similarities and the differences are highlighted in this section.

Address entries within an ACL, which is used to filter users who are affected by the ACL, often contain wildcard characters (*). However, ACIs cannot use wildcard characters in their O/R Address entries. Any valid O/R Address attribute and value can be specified. Matches are found when:

- A user's address and the address in the ACI are identical.
- The address in the ACI is an exact subset of a user's address.

For example, this address within an ACI:

```
S=Sand/OU2=sales
```

is a subset of, and therefore matches, the following user address:

```
G=George/S=Sand/OU1=paris/OU2=sales
```

However, it does not match the user address:

```
G=George/S=Sand/OU1=paris/OU2=marketing
```

because there is a mismatch between OU2=sales and OU2=marketing.

Group entries within an ACI follow the same conventions as group entries within an ACL:

- The local administrator group's default permissions are config, remove, modify, and read.
- The local user group's default permissions are remove, modify, and read.
- The default group's default permission is read.

PDL Commands

The following table lists and describes commands associated with Public Distribution Lists (PDLs), including commands for manipulating Access Control Information.

If a PDL has an associated ACI, the user must have certain permissions defined in the ACI to use PDL commands. <Xref_Color>"ACI Permissions for PDL and Directory Commands" for more information.

Command	Description
omaddpdl	Add a Public Distribution List
omaddpdln	Add an entry to a Public Distribution List
omdelpdl	Delete one or more Public Distribution Lists
omdelpdln	Delete one or more entries from a Public Distribution List
ommodpdl	Modify a Public Distribution List
ommodpdln	Modify Public Distribution List entries
omshowpdl	List Public Distribution Lists
omshowpdln	List entries in a Public Distribution List
omaddaci	Add an Access Control Information member

Command	Description
omchkaci	Check Access Control Information permissions for a user
omdelaci	Delete an Access Control Information member
ommodaci	Modify an Access Control Information member
omshowaci	Show the contents of Access Control Information

ACI Permissions for PDL and Directory Commands

The user must have the required ACL permissions to access the specified PDL or Directory.

If a PDL has an associated ACI, a user also must have the following permissions defined in the ACI to the use PDL commands, as listed in the following table.

PDL Command	ACI Permission
omaddpdl	none
omaddpdln	modify and read
omdelpdl	remove and read
omdelpdln	modify and read
ommodpdl	modify and read
ommodpdln	modify and read
omshowpdl	read
omshowpdln	read

If a PDL has an associated ACI, a user also must have the following permissions defined in the ACI to the use Directory commands, as listed in the following table.

Directory Command	ACI Permission
omaddent	none
omdelent	remove
ommodent	modify
omsearch	read

Note The Linux root user, and the owner of the Directory, always has all permissions. You must use ACI commands of the type om*aci, when manipulating ACIs. You must not use normal Directory commands (that is, omaddent, omdelent, ommodent and omsearch) for this purpose. For more information about Directory commands, see “CDA Command Summary” on page 100.

Chapter 11

Public Folders

The contents of this chapter have been organized into the following sections:

Note	Public folders can be accessed only by Premium users. For more information, see "About Scalix Product Editions".

Public Folder Overview

Public Folders are shared areas in the Message Store. They enable work groups to share information.

Public Folders can contain the following items:

- Basic items (for example, text, distribution lists, word-processing documents)
- Messages
- Other (“nested”) Public Folders

Public Folders can be set up with Access Control Lists to restrict access to certain users or classes of user. In addition, expiry dates can be set so that short-lived information is automatically deleted after a specified period.

By default, only Scalix Administrators can create top-level Public Folders. Users can create nested Public Folders within these top-level Public Folders.

Note	Scalix commands and directories use the term “Bulletin Board” and the abbreviation “bb” to refer to Public Folders.

Client Access to Public Folders

Access to Public Folder functionality through Scalix clients depends on the permissions of the clients. The UAL_DISABLE_BB option in the **general.cfg** file enables administrators to deny or allow access to Public Folders from UAL clients.

Users can add items to Public Folders by cutting and pasting, dragging and dropping, sending mail messages to the Public Folder, and so on.

How Messages Are Mailed to Public Folders

Usually, items are added to Public Folders directly through the Client Interface. However, you can attach items to Public Folders by mailing messages to the BB Server.

To mail a message to a Public Folder for which you have access, address the message to any mailnode on the system, using a Personal Name of USER +BB. Identify the required Public Folder by specifying the subject of the Public Folder in the DDA of the address. For nested Public Folders, enter the full pathname of the Public Folder. Each Public Folder subject is separated by the "greater than" symbol (>).

Ownership of Public Folder Items

Each Public Folder item has a Creator and an Attacher. The Creator is the Scalix user who created the item. The Attacher is the Scalix user who added the item to the Public Folder.

Both the Creator and the Attacher have full access permissions for their associated item. That is, both of them can delete or modify the item. This permission is independent of any entries in the Public Folder's ACL (see "Adding a User to a Public Folder ACL" on page 199).

Any user who modifies a Public Folder item becomes its Creator.

For example, User_A creates a file. User_B copies it and adds it to a Public Folder. The Public Folder shows User_A as the Creator of the item, and User_B as the Attacher of the item. Both User_A and User_B can modify or delete the item.

If User_A subsequently modifies the item, they remain as the Creator, and so both User_A and User_B retain the ability to modify or delete the item.

However, if User_B modifies the item, they become the Creator. In this case, only User_B, as both Creator and Attacher, can make any subsequent changes to the item (although a user with the delete access permission can still delete the item).

Status of Public Folder Items

When you access a Public Folder, your Scalix client can highlight any items that were attached since the last time you accessed it. These new items remain highlighted until you read them. Scalix uses sequence numbers to provide this feature.

Whenever an item is added to a Public Folder, Scalix assigns it a unique number to identify it. These numbers are assigned sequentially as new items are added, and are not reused.

Scalix stores sequencing information in the two files listed in the following table.

File Name	Description
~scalix/bb/sys.seq	This is the system sequence file. Each Public Folder has an entry in this file. The highest sequence number currently assigned within each Public Folder is stored within in the appropriate entry.
~scalix/user/g..../000002m	This is the user sequence file. Every Public Folder accessed by the user has an entry in this file. The highest sequence number present when they last accessed each Public Folder is stored within the appropriate entry.

Using Public Folders

The following information describes how to manage Public Folders.

Listing Top-level Public Folders

You can list the top-level Public Folders using the command:

```
omlistbbs
```

Adding a Top-level Public Folder

You can add a top-level Public Folder using the command:

```
omaddbb -s subject
```

Creating Nested Public Folders

A nested Public Folder is a Public Folder that is attached to another Public Folder. A nested Public Folder can be attached to a top-level Public Folder or to another nested Public Folder.

By default, only Scalix Administrators (or the user "root") can create top-level Public Folders. Other users can create nested Public Folders. However, you can prevent clients from creating any nested Public Folders, by setting the general configuration option UAL_DISABLE_NESTED_BBS to TRUE. See "Configuration Options" on page 301 for more information.

Use the -B, -m, -n and -d options with the omaddbb command to create nested Public Folders. The following table lists the ways you can add a nested Public Folder.

Command	Description
omaddbb -m Sales -s UK	This command creates a nested Public Folder called UK within the Sales Public Folder using the full pathname.

Command	Description
omaddbb -n 1 -s UK	This command creates a nested Public Folder called UK within the Sales Public Folder using the Public Folder's temporary reference number.
omaddbb -B 1249 -s UK	This command creates a nested Public Folder called UK within the Sales Public Folder using the Public Folder's absolute reference number.

Attaching a Message to a Top-level Public Folder

To attach a message to the top-level Public Folder named Announcements, where one of the mailnodes on the system is OU1=ny/OU2=sales/OU3=mis, address the message to:

```
USER +BB/ny,sales,mis(Announcements)
```

Attaching a Message to a Nested Public Folder

To attach a message to a nested Public Folder named Social which is in the Public Folder Announcements, which a top-level Public Folder, address the message to:

```
USER +BB/ny,sales,mis(Announcements>Social)
```

Note You must enable the Public Folder Server before it can process any messages (omon -s bb). See "Public Folder Server" on page 150 for more information.

You can use all four DDAs if the pathname of a nested Public Folder is long. The DDAs are concatenated. and the maximum length is 1 Kbyte. If a Public Folder subject contains a > character, it must be "escaped" by preceding it with a backward slash (\). If there is more than one Public Folder with the same pathname, the first is used.

Generally, the Public Folder commands assume you are working in a top-level Public Folder. To work with nested Public Folders, remember to specify where in the hierarchy you wish to work. Use the -d option with the Public Folder commands to specify the depth at which you wish to work. For example, the omlistbbs -d 2 command displays the top-level Public Folder and the next level down.

Maintaining Public Folders

Scalix allows you to add and delete Public Folders, and add or delete ACL entries for Public Folders. You can use the ommaintbb command to remove old Public Folder postings. You can clean out all Public Folders or just one. If you are cleaning out a specific Public Folder, this command works recursively.

The following table lists the basic ommaintbb command options that allow you to delete Public Folders.

Option	Description
-a	Use this option to delete items that have reached their expiration date.
-e	Use this option to delete items that are older than the number of days you specified.

Option	Description
-I	Use this option to delete all items imported through a synchronization agreement.

Reference Numbers

Scalix Public Folders are assigned reference numbers. The following sections describe the type of numbers assigned:

- “Temporary Numbers”
- “Permanent Numbers”

Temporary Numbers

Newly created Public Folders are assigned temporary reference numbers. These numbers can change as more Public Folders are added or deleted. An example of temporary numbering follows:

- 0: PUBLIC FOLDER AREA
- 1: Sales
- 2: Marketing

Use the omlistbbs command with no option to view Public Folders with their temporary reference numbers.

Permanent Numbers

Scalix also assigns permanent reference numbers to Public Folders. These are called absolute reference numbers and are never reused by Scalix. An example of absolute numbering follows:

```
1057: PUBLIC FOLDER AREA
1249: Sales
2825: Marketing
```

Use the omlistbbs command with the -b option to view Public Folders with their absolute reference numbers.

To display the absolute reference numbers of nested Public Folders, enter a command similar to the following:

```
omlistbbs -d 2 -b
```

This commands displays two levels of Public Folders. To view the entire structure of the Public Folder, use the -d 0 option.

Expiry Dates and Expiry Delays

Expiry dates and expiry delays enable out-of-date information to be removed from Public Folders automatically. You set up an expiry date or expiry delay when you create the Public Folder.

Expiry Dates

Set an expiry date if you want all the items attached to the Public Folder to be deleted when that date is reached. Setting an expiry date overrides any expiry delay value set for the Public Folder. By default, a nested Public Folder inherits the expiry date of its parent, but you can specify a different date when you create it.

Set an expiry date by using the omaddbb -t command.

Expiry Delays

Set an expiry delay if you want each item attached to the Public Folder to be deleted when it reaches a certain age (in days).

Set an expiry delay by using the omaddbb -l command. Include the -S option if you want to allow users to override this, and set their own expiry delays for some individual items.

You can prevent users from setting expiry delays for individual items by including the -U option with the omaddbb command.

Public Folder Server

The Public Folder Server (BB Server) is used to synchronize Public Folders and attach messages mailed to it to the appropriate Public Folders (some clients mail messages to Public Folders rather than attaching them directly through the Client Interface).

The following table lists key Public Folder Server information:

Item	Name
Service name	bulletin
Process name	bb.server
Queue name	BB
Given name	message-type
Surname	+BB

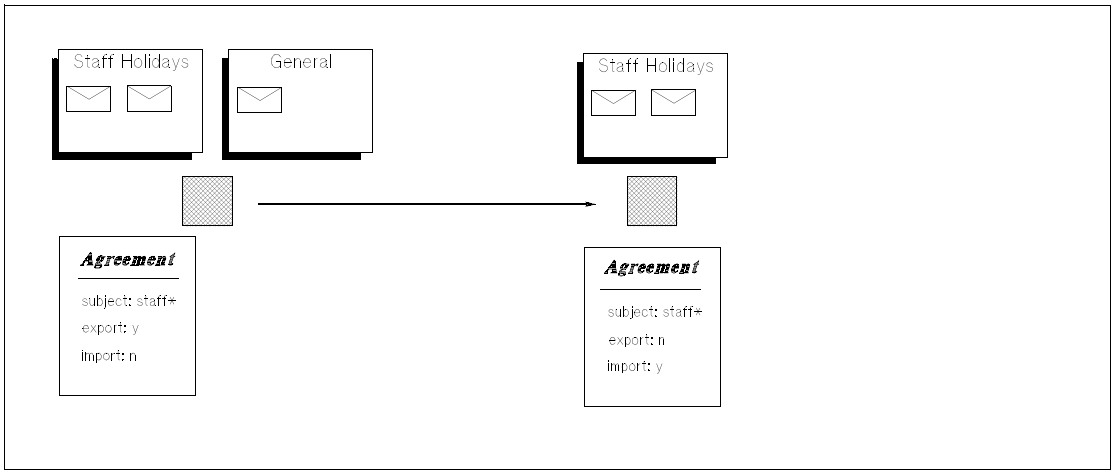
Public Folder synchronization is the process of automatically updating Public Folders and their contents from one system to another. The process ensures that when you add an item to one Public Folder, the same item is also added to all equivalent Public Folders in the network.

Synchronization is managed using Public Folder synchronization agreements. These define the rules of each exchange. Each agreement defines whether items are imported or exported, and the Public Folders to which the agreement applies. All items within the hierarchy of a specified Public Folder are included in the agreement.

Synchronization is performed by exchanging mail messages between two Public Folder Servers (BB Servers). Each message adds one item to a Public Folder. An item is anything that can be added to a Public Folder, such as messages, other Public Folders, or files. The "sending" server is the exporting BB server, the "receiving" server is the importing BB Server.

Synchronization Agreements

The following illustration shows a Synchronization Agreement.



Matching agreements must exist on the exporting and importing systems before items can be exchanged. Typically, a number of agreements are specified on each system, with each agreement specifying the exchange for several Public Folders.

Synchronization Concepts

Public Folder synchronization is used to replicate information across a number of servers in your network. It ensures that the same information is available across a number of system and reduces network traffic.

For example, a corporation in New York has a server (A) that has a Public Folder called Sales. This contains attachments (mail messages, text files, distribution lists, and so on) that relate to the Sales function of the corporation. To give the branch office in Los Angeles access to the information on the Sales Public Folder, you set up an export synchronization agreement to export the Sales Public Folder to the Los Angeles server (B). A corresponding import agreement must also be set up on Server B.

This creates a Public Folder on Server B named Sales that contains exactly the same attachments as the Sales Public Folder on Server A. At regular intervals, updates to the Public Folder on Server A are automatically sent to the Public Folder on Server B to ensure that the two Public Folders are synchronized.

The items on Sales on Server A are master items, because they were originally created on Server A. The same items on Sales on Server B are slave items, because they are copies of the master items. Whether an item is a master or a slave is important when deleting or modifying items.

Users on Server B can also attach items to their version of Sales. However, these are not replicated on Server A until you create the appropriate export synchronization agreement on B and a corresponding import agreement on A. When you do this, the two Public Folders become identical, except that master items on A are slave items on B, and slave items on A are master items on B.

You specify whether object files attached to messages or items are included when the message or item is exchanged during synchronization. This is set using the BBS_SEND_OBJECT_FILES option in the general configuration file (~/sys/general.cfg). See “Configuration Options” on page 301 for more information.

Deletion of Public Folders

Despite synchronization agreements, the deletion of Public Folders is not replicated across servers. In the example above, if you delete the Sales Public Folder on Server A, the Sales Public Folder on Server B is not deleted.

Renaming of Public Folders

Renamed Public Folders are replicated across servers. If you have the appropriate access rights, you can modify the name of a Public Folder and replicate this modification to copies of the Public Folder that exist on other servers, provided that the appropriate synchronization agreements exist.

This propagation occurs whether or not you modify the original Public Folder or one of its copies.

Public Folder Synchronization and Folders

While folders are similar to Public Folders, they behave differently under synchronization:

- Users cannot add any items to folders that are slave items.
- When a user adds an item to a folder that is a master item, the entire contents of the folder is replicated to all synchronized Public Folders.

Caution	For the above reasons, Scalix recommends not attaching folders to Public Folders, either directly or to other Public Folder items. Scalix also recommends replacing any existing folders within Public Folders by nested Public Folders to ensure that Public Folder synchronization operates correctly. To replace a folder with a nested Public Folder, create the nested Public Folder, and copy or move the individual items from the folder to the Public Folder. You can then delete the folder.

Modification of Public Folder Items

Modification of a Public Folder item includes one or more of the following actions:

- Changing the content of the item

- Changing the name of the item
- Adding, removing, or changing an attachment to the item

You can modify a master item, provided you have the required permissions. However, you cannot modify a slave item.

To simulate modification of a slave item, you can copy it, and then delete it from the Public Folder. Edit it as required and attach the edited copy to the Public Folder.

Addition of Public Folder Items

Any top-level Public Folder items that are created replicated to other servers provided that the relevant import and export synchronization agreements exist. These items are master items on the server on which they were created, and slave items on the servers to which they are replicated.

Adding an attachment to a Public Folder item is considered as a modification by the synchronization process. Therefore, you can add attachments to master items (although this results in the replication of the entire original item with all of its attachments), but not to slave items.

For example, you can attach a text file or a message to an existing folder on a Public Folder on Server A, provided that the existing folder was originally created on Server A (and provided you have the required permissions). The synchronization process detects this addition as a modification of the original top-level, folder, and causes the entire folder to be replicated to the other servers.

Deletion of Public Folder Items

Deleting a master item result in the deletion of all its slave items. However, deleting a slave item does not delete the master item or any other similar slave items.

For example, a user on Server A adds a mail message to the Sales Public Folder on Server A, and this is copied, as a result of synchronization agreements, to the Sales Public Folders on Server B and Server C. A user on Server B can delete this item (provided they have the required permissions), but the message remains on the Public Folders on Server A and Server C.

If a user on Server A deletes the item, the item is deleted from the Public Folders on Server B and Server C.

You can change this default behavior by setting two general configuration options:

- BBS_PROPAGATE_SLAVE_DELETION

  When set to TRUE, this causes deletion of a slave item to be replicated to all other slave copies of the item, and also to the master item if the BBS_DELETE_MASTER_BY_SYNC option is set to TRUE.

- BBS_DELETE_MASTER_BY_SYNC

  When set to TRUE, this allows a master item to be deleted when one of its slave copies is deleted. BBS_PROPAGATE_SLAVE_DELETION must also be set to TRUE for this to occur.

See “Configuration Options” on page 301 for more information.

Deletion of Slave Item Attachments

You cannot delete an attachment from a slave item. For example, if a Public Folder slave item has an attached folder, which in turn contains several text messages, you cannot delete any of these messages, or the folder itself. This is because the synchronization process considers this to be a modification of the top-level item and modification of slave items is not allowed.

The Synchronization Process

The synchronization process is configured using the commands omaddbbsa, omdelbbsa, ommodbbsa, and omlistbbsa.

When the BB Server is started, it reads the Public Folder synchronization agreements contained in the file ~/sys/bb/sync.conf. If any agreements are added, deleted, or modified, you must stop and start the BB Server before the change takes effect.

The BB Server operates in two modes; import mode and export mode. It begins in import mode and processes messages waiting on the BB queue. When it finishes processing incoming messages, it switches to export mode and generates messages for export. When it finishes generating messages for export, it immediately switches back to import mode. If no messages are waiting on the BB queue, it waits in import mode until a timer expires and then switches to export mode. The timer defaults to 60 minutes. See “Configuration Options” on page 301 to change the default timer value using the BBS_CUST_CHECK_TIME option.

The BB Server in Export Mode

In export mode, for a specific export agreement, the BB Server searches the Public Folder area for items that have been added after the export agreement was last processed. The time is stored in a sequencing file in ~/bb.

Each new item is sent to the importing BB Server. The message consists of two body parts; the first contains information about the agreement in IA5 text (including where the item belongs in the Public Folder area), and the second contains the item itself. Each message contains one item only.

The information describing where the item belongs on the Public Folder includes flags for each top-level and nested Public Folder under which the item is held. The flag autoadd if set to YES means “create the Public Folder on the importing system if it does not already exist”. The flag is set to YES if, when the exporting BB Server begins processing in export mode, it cannot find an existing sequencing file for that agreement; that is, this is the first time this agreement has been processed. If the Public Folder is later deleted from the importing system, it is not recreated by incoming synchronization messages from the current agreement as the flag autoadd is be set to NO.

To prevent looping of items between systems, the agreement number associated with the item is checked and the message is given an ID.

- The agreement number (the number shown in the omlistbbsa command and added to the item when it is imported), is checked. If it exists and is the same as the agreement number of the exporting agreement, the item is not exported.

- The message ID is added to a message when it is exported (if it does not already exist). The ID contains the O/R Address of the exporting system. The message ID is used by importing BB Servers to check the message did not originate on the importing system.

The BB Server in Import Mode

In import mode, the BB Server reads messages from the BB queue. If the given name in the recipient address is OMSYNC, the BB Server assumes the message contains an item to be synchronized.

The importing BB Server checks the O/R Address in the message ID of the incoming message does not match its own O/R Address. If it does, the message is considered to be looping, and is discarded.

The server adds the item to the appropriate Public Folder using the information contained in the first body part of the message. If an item of the same name in the same location already exists and its message ID matches that of the new item, the new item is discarded.

When the item is added, the BB Server also adds the agreement number (the number shown in the omlistbbsa command) to the item. The agreement number is used by the exporting BB Server to prevent items being sent back to the system from which they were imported.

If an error is detected by the BB Server while it is processing the message, the message is sent to the Error Manager for the importing system.

The importing BB Server does not return any kind of acknowledgment to the exporting server. If the item is not added, the BB Server records it in the Event Log of the importing system.

While a Public Folder exists, the BB Server records any instance of a message for the Public Folder not being used in the Event Log. For example, when the same message is sent by two different exporting BB Servers.

Access Capabilities Required for Public Folder Synchronization

The following minimum-level access capabilities are required to import items:

- The originator of the message (the exporting BB Server) has use access on the importing Public Folder Server
- The originator of the message (the exporting BB Server) has read access to the Public Folder area, and attach and delete access to the Public Folder being synchronized.

  If the top-level Public Folder does not already exist, then the originator of the message must have attach and delete access to the Public Folder area.

Synchronization messages (type OMSYNC) are automatically given the capabilities of the admin and default groups in addition to any permissions explicitly granted in the ACL to the O/R Address pattern of the originator.

The minimum access permissions are required to export items are the combined permissions of the standard groups local, default and admin plus any permissions explicitly granted in the ACL to the O/R Address pattern of the originator
(OMSYNC +BB/local_primary_mailnode) must give read access to the Public Folder area and the Public Folders being synchronized.

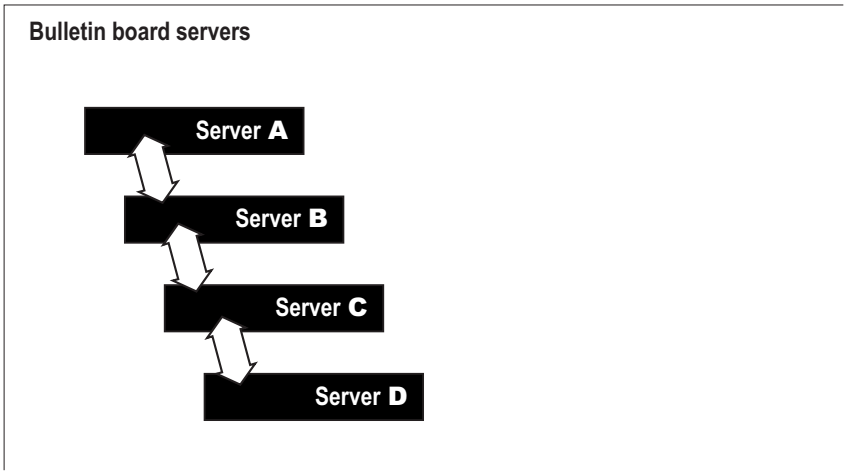
The Public Folder ACLs themselves are not exported when Public Folders are synchronized. New top-level Public Folders created as a result of a synchronization agreement use the default Public Folder ACL settings. New nested Public Folders created as a result of a synchronization agreement inherit their ACLs from their parent Public Folder.

Synchronization Topologies

Public Folders can be synchronized in chains, where each BB Server passes new items to the next Bulletin Board (BB) Server in the chain, or synchronized in a hub system, where every BB Server receives updates from one central server.

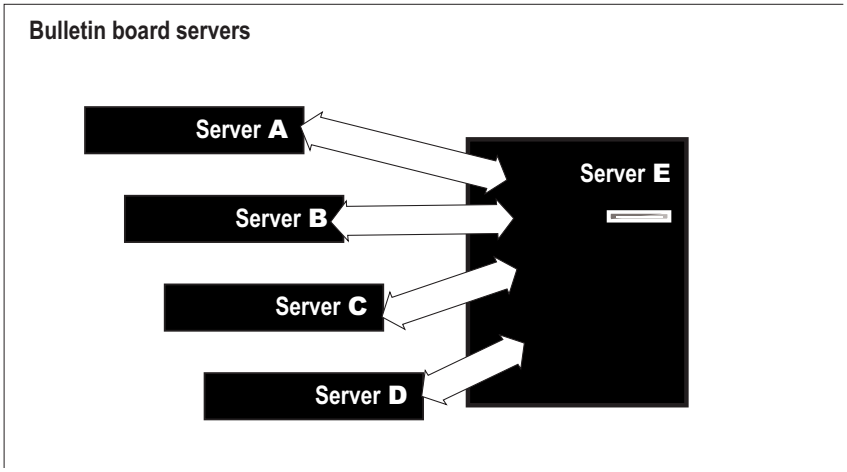
Synchronization Chains

An example synchronization chain is shown in the following illustration. In this case, a user can add an item to a Public Folder on Server A. This item will then be replicated, provided the relevant synchronization agreements exist, to the equivalent Public Folder on Server B, and then to Servers C and D.

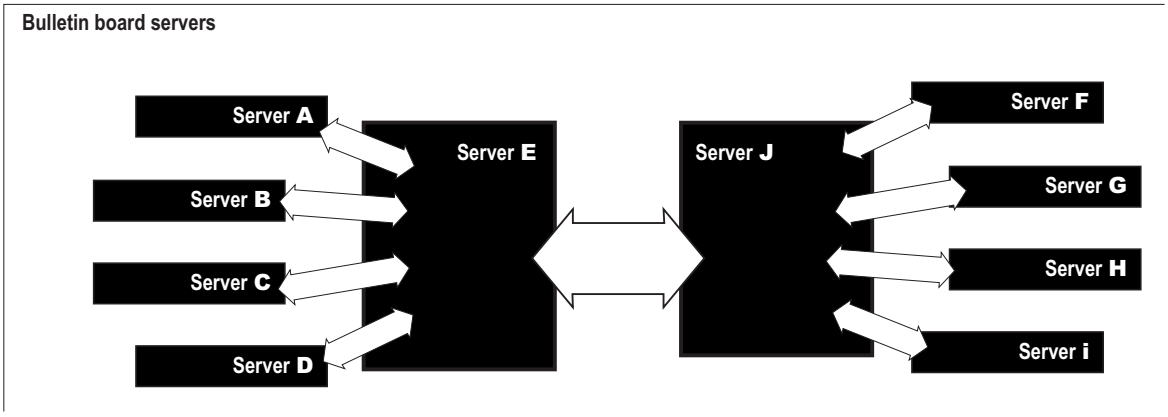


Synchronization Hubs

The following illustration shows a synchronization hub, in which all Public Folder amendments are sent to Server E, which then propagates them to the other servers. This configuration can be extended to a linked hub, as shown in the following illustration.



This configuration can be extended to a linked hub (Server J), as shown in the following illustration.



Other Synchronization Topologies

There are a number of other synchronization topologies. Some of these, however, are complex to administer, could cause network congestion, and might not give the required results.

For example, grids (where all systems are updated by each other) and loops (linked chains) can result in duplication of synchronization messages, and lead to a loss of synchronization. Scalix recommends simpler Public Folder Synchronization agreements.

Synchronization Agreement Guidelines

Note the following guidelines when setting up Public Folder synchronization agreements:

- Always activate the importing synchronization agreement before activating the corresponding remote exporting agreement. This avoids exported items arriving at the importing system before the importing system can accept them. If this occurs, the items are discarded and the event is recorded in the Event Log.

  If the agreement is to both import and export, make sure the corresponding agreement on the remote system is activated at the same time.
- If an export agreement is deleted and then added again, it will behave as if it is a completely new agreement. This is because the agreement numbers for the old and new agreements (as shown by omlistbbsa) will be different.
- If the primary mailnode on a system is changed, all agreements (on both importing and exporting systems) must be changed to reflect the name of the new primary mailnode.

  As a result of changing the primary mailnode in the agreements, new message IDs are generated for all items being exported. Consequently, the same item, differing only in message ID, will be imported, resulting in duplicate items.
- Use the wildcard character (*) when specifying the subjects of top-level Public Folders to import or export. This avoids adding individual agreements for each Public Folder. However, beware that indiscriminate use can lead to significantly increased network traffic.

  Wildcard characters represent zero or more characters. One or more wildcard characters can be placed anywhere in the subject string.

Note	If using a multibyte character set, wildcards can be placed only at the beginning and end of the subject string. Also, the output from the Scalix commands displays multi-byte characters as asterisks, so users cannot distinguish between a subject containing wildcards and one containing multibyte characters.

- To ensure specific Public Folder s cannot be exported or imported, no matter what agreements are set up, restrict the access capabilities granted in the Access Control Lists (ACLs) for the Public Folder area and for specific Public Folders.

Messages Sent to the Public Folder Server

"USER" type messages are received by the Public Folder Server. The Public Folder Server attaches a message addressed to it with the given name of USER to the Public Folder specified in the DDA of the address.

Messages are sent to the Public Folder server as follow:

1. Messages addressed to the Public Folder Server are routed in the normal way by the Service Router.

   If the message is for a local address, it is passed to the Local Delivery Service by the Service Router.
2. The Local Delivery Service attaches messages addressed to the entity +BB to the Public Folder Server queue BB.
3. Public Folder Server adds the message to the correct Bulletin Board.

Public Folder Commands

The following table lists commands associated with Public Folders.

Command	Description
omaddbb	Add a top-level Public Folder
omdelbb	Delete a top-level Public Folder
omlistbbs	List top-level Public Folders
ommaintbb	Maintain top-level Public Folders
ommodbb	Modify the subject of a top-level Public Folder
omshowbb	Show details of a top-level Public Folder
omaddbbsa	Add a Public Folder Synchronization agreement
omdelbbsa	Delete a Public Folder Synchronization agreement
omlistbbsa	List Public Folder Synchronization agreements
ommodbbsa	Modify a Public Folder Synchronization agreement

Chapter 12

Internet Address Creation

Local users and Public Distribution Lists (PDLs) can be given Internet addresses, in addition to their Scalix addresses. The Internet Mail Gateway can then map between Scalix addresses and their corresponding Internet addresses. The Internet addresses that are created are stored as the INTERNET-ADDR attribute (also known as IA). You can create these Internet addresses manually or you can configure automatic Internet address generation.

This chapter describes both methods of creating Internet addresses for users and PDLs.

Automatic Generation of Internet Addresses

To generate Internet addresses automatically for local users and PDLs, you first associate an Internet domain with a mailnode. When you subsequently create users or PDLs on that mailnode, their Internet addresses are automatically generated based on the Internet Domain of their mailnode and their Scalix name attributes.

There are two methods of associating an Internet domain with a mailnode:

- By explicitly specifying an Internet Domain for each mailnode.

  This mechanism is described in "Specifying Internet Domains for Mailnodes" on page 163.

- By specifying rules by which an Internet domain is created based on an Scalix address pattern.

  When you subsequently create a mailnode that matches the address pattern, an Internet domain is automatically associated with it. As described above, when you create users or PDLs on that mailnode, their Internet addresses are automatically generated.

This method provides a further level of automation, and is useful for systems with a large number of mailnodes, or those with a rapidly changing set of mailnodes.

This mechanism is described in "Configuring Internet Addressing for Scalix Address Patterns" on page 164.

Uniqueness of Internet Addresses

The Internet addresses of local users must be unique among all local users and, if replicated in the System Directory, unique among all entries in the System Directory. Similarly, the Internet addresses of PDLs must be unique in the Directory in which they exist. Uniqueness checking for all Directories is automatic.

You can use the following configuration options to override this default:

```
DIR_IA_UNIQUECHECK_ON

DIR_IA_UNIQUECHECK_OFF
```

The value of each of these options is a comma-separated list of those directories for which uniqueness checking of Internet addresses is turned on and off. See "Configuration Options" on page 301 for more information.

Turning On Automatic Internet Address Generation

You can automatically generate Internet addresses when the configuration option INET_USE_AUTO_IAM is set to TRUE. This allows you to use the commands described in "Specifying Internet Domains for Mailnodes" on page 163 and "Configuring Internet Addressing for Scalix Address Patterns" on page 164 to configure Internet addresses or domains.

If automatic Internet address generation is turned off, using these commands to configure Internet addresses will generate errors.

Note	Turning on automatic Internet address generation also turns on automatic mapping between Internet addresses and Scalix addresses at the Internet Mail Gateway. You can turn this automatic mapping off by setting the configuration options UXO_NAME_MAPPING, UXI_NAME_MAPPING, and BRW_NAME_MAPPING to FALSE. Beware, however, that turning mapping off could cause messages to be routed incorrectly. See "Configuration Options" on page 301 and "Scalix Interfaces and Gateways" on page 25 for more information.

Specifying Internet Domains for Mailnodes

You can associate an Internet domain with a mailnode. The Internet domain is used subsequently, when you create users or PDLs on that mailnode, to generate Internet addresses for those users or PDLs.

The Internet domains you supply using the commands described below are stored in the mappings file ~sys/ornia.

Use the omshowmn -m mailnode command to see the mappings you have set up.

Configuring an Internet Domain When You Create a Mailnode

Use one of the following forms of omaddmn to set an Internet domain for a mailnode:

```
omaddmn -m mailnode -D domainname [-N name]
```

domainname is the name of the domain that will be associated with mailnode. name is either a simple rule to specify how the name part of the Internet address will be formed (for example, ${G}.${S} to produce givenname.surname) or the name of a script that generates the name part of the Internet address (see "Scripts for Internet Address Generation"). If the -N option is not specified, a default of ${G}.${I}.${S} is used.

Note	**name** can be a script that generates the entire Internet address, including the domain. In this case, although you must specify a value for domainname, it is redundant.

```
omaddmn -m mailnode -Q
```

Use this option to specify that no Internet mail address will be generated for the mailnode.

If you use the omaddmn command without the -D or -Q options, Scalix will attempt to determine the mailnode's associated Internet domain from existing information in the ~sys/ornia file (see "Storing Internet Addressing Information" on page 166), and will generate a warning if it cannot do so.

Configuring an Internet Domain When You Modify a Mailnode

You can use the -D and -N options, described above, with the ommodmn command to specify the Internet domain for a mailnode. If you do not specify these options then, by default, ommodmn does not attempt to generate or modify an Internet domain.

You can force ommodmn to generate an Internet domain by using the -A option:

```
ommodmn -m mailnode -A
```

In this case, Scalix will attempt to determine the mailnode's associated Internet domain from existing information in the ~sys/ornia file (see "Storing Internet Addressing Information" on page 166).

You can change the default behavior of ommodmn by setting the configuration option INET_AUTOGEN_IA_ON_MODIFY to TRUE (see "Configuration Options" on page 301 for more information). This is equivalent to using the -A option.

Examples

- omaddmn -m 'sales,us' -D 'sales.loc1.org.com'
  omaddu -n 'John T. Smith/sales,us'

  John T. Smith/sales,us

  will have an Internet address of:

  John.T.Smith@sales.loc1.org.com.

- omaddmn -m 'research,us' -D 'research.loc2.org.com' -N '${G}_${S}'
  omaddu -n 'Catherine Jones/research,us'

  Catherine Jones/research,us

  will have an Internet address of:

  Catherine_Jones@research.loc2.org.com.

Configuring Internet Addressing for Scalix Address Patterns

Rather than explicitly specify Internet addressing information for each mailnode, as described in "Specifying Internet Domains for Mailnodes", you can specify how such information is to be constructed. This is stored in the ~sys/ornia file, as described in "Storing Internet Addressing Information" and used as follows:

- When you create a mailnode that matches an address pattern in ~sys/ornia.

  The corresponding Internet domain is calculated and associated with the mailnode.

- When you create a Directory entry that matches an address pattern in ~sys/ornia and that does not correspond to a local user or PDL.

  An Internet address for the Directory entry is calculated and assigned to the entry's INTERNET-ADDR attribute.

Use the omaddiam and ommodiam commands as follows to specify Internet addressing information for an Scalix address pattern.

```
omaddiam -m OM_address -D domain -N name
```

OM_address is the Scalix address pattern, excluding Personal Name attributes, to be associated with an Internet address. It can include wildcards, for example, ny,sales,*.

domain can be the name of the domain that will be associated with OM_address, a simple rule (for example, ${OU2}.abc.def.com), or the name of a script (see "Scripts for Internet Address Generation"). name specifies how the name part of the Internet mail address will be formed, and is either a simple rule (for example, ${G}.${S}) or the name of a script. If the -N option is not specified, a default of ${G}.${I}.${S} is used.

Note	name can be a script that generates the entire Internet address, including the domain. In this case, although you must specify a value for domain, it is redundant.

To modify existing Internet address configurations, use the ommodiam command, with the same options as omaddiam.

To delete Internet address configurations, use the omdeliam command.

Use the omshowiam command to see the Internet address configurations you have set up.

Example

Enter the following command to set the Internet addressing information for all mailnodes that match the pattern *,*,*:

```
omaddiam -m '*,*,*' -D '${OU3}.${OU2}.${OU1}.com' -N '${G}.${S}'
```

When you create the mailnode ny,bigco,sales, it will automatically be associated with the domain sales.bigco.ny.com. Create the mailnode in the normal way:

```
omaddmn -m 'ny,bigco,sales'
```

Now, when you create the user Chris Wolf/ny,bigco,sales, his Internet address is automatically configured to be Chris.Wolf@sales.bigco.ny.com:

```
omaddu -n "Chris Wolf/ny,bigco,sales"
```

Creating Internet Addresses for non-Scalix Users

You can use the omaddiam command to generate Internet addresses automatically for non-Scalix users. When you add these users to the Scalix Directory, using the omaddent command, an INTERNET-ADDR attribute is automatically created for them.

For example, if you regularly correspond with employees of OtherCo Corp., whose Internet addresses are of the form firstname.lastname@otherco.com, enter the following command:

```
omaddiam -m '*/otherco' -D '$(OU1).com' -N '$(G).$(S)'
```

Now, enter these OtherCo employees in the Scalix Directory. For example:

```
omaddent -a -e s=Joe/g=Smith/ou1=mailnode
```

Joe Smith will be added to the Directory with an INTERNET-ADDR attribute of Joe.Smith@otherco.com.

As with creating and modifying mailnodes, you can use the -Q option with omaddent to prevent an automatic Internet address from being generated, and the -A option with ommodent to force an Internet address to be generated. See the relevant man pages for more information.

Testing and Previewing Internet Address Configurations

You can use the omaddiam, ommodiam, omdeliam, and omshowiam commands even when automatic Internet address generation is disabled.

This allows you to test your Internet addressing configuration before implementing it. Use the ompreviewia command to see the effects of your configuration. See the man page for ompreviewia for more information.

Storing Internet Addressing Information

The Internet addressing information you supply using the omaddmn, ommodmn, omaddiam, and ommodiam commands are stored in the file ~sys/ornia. This file is checked when you create users/PDLs and when you create mailnodes:

- When you create a mailnode, a check is made to see if it matches an address-pattern entry in the file. If a match is found, the Internet domain is calculated for the mailnode, and a new entry is placed in the ~sys/ornia file for the mailnode.
- When you create a user/PDL, a check is made to see if their mailnode matches a mailnode entry in the file. If a match is found, their Internet address is calculated and assigned to their INTERNET-ADDR attribute.

For example, enter the following command:

```
omaddiam -m '*,*' -D '${OU2}.${OU1}.com'
```

Now create the mailnode uk,sales:

```
omaddmn -m 'uk,sales'
```

Because this mailnode matches the address pattern *,*, an associated Internet domain is calculated according to the omaddiam command entered earlier. This is sales.uk.com. An entry that associates the mailnode uk,sales with the Internet domain sales.uk.com is now placed in the file ~sys/ornia.

Use the omshowmn command to confirm the Internet domain associated with uk,sales:

```
omshowmn -m uk,sales
```

This domain is used to construct an Internet address for users and PDLs created subsequently on mailnode uk,sales.

If you now use the ommodiam command to change the entry for *,* in ~/sys/ornia, the change will only affect any subsequent mailnodes that you create. The entry in ~sys/ornia for uk,sales is not affected.

Updating Address Configurations

The omgeniamods command assists you to update users' Internet addresses to reflect the automatic settings you have configured. For example, you could use this command if you have upgraded from release B.06 to B.07, and wish to take advantage of automatic Internet address generation. Also, you could use the command if you change the Internet addressing information for a mailnode, and wish to update the relevant users to reflect the new information.

Before running the command, you must decide if uniqueness of Internet addresses is required (see “Uniqueness of Internet Addresses”). If it is, use the omchkattuniq command to check for duplicate Internet address entries.

The omgeniamods command generates a script that you can edit as required and then run to amend the Internet addresses for users. See the man page for omgeniamods for more information.

Scripts for Internet Address Generation

Scripts that are specified in the omaddmn and omaddiam commands, to generate all or part of an Internet address, must be placed in ~/rules. A sample script is provided:

```
/opt/scalix/examples/general/ornia.map
```

This script describes and demonstrates the protocol that scripts must follow. You can use this as a template to develop your own scripts.

Manually Generating Internet Addresses (Users and PDLs)

When you create a user or a PDL, their Internet address is automatically configured, together with one for each alias, provided the mailnode has an associated Internet domain.

However, you can configure an Internet address directly, if an automatic value cannot be generated or if you wish to override the automatic value (because, for example, the automatic value would not be unique).

The following sections describe the commands you use to configure an Internet address for a user or PDL.

Creating an Internet Address When You Add a User

By default, the omaddu command causes an Internet address attribute to be generated for the user you add, providing their mailnode is configured appropriately. You can change this behavior by using one of the following forms of the command:

```
omaddu -n name/mailnode/internet-addr=internet_address
```

internet_address is the full Internet address of the user, and will override the automatically-generated Internet address.

```
omaddu -n name/mailnode -Q
```

Use the -Q option to prevent an Internet address from being automatically generated for this user.

Creating an Internet Address When You Modify a User

By default, ommodu does not automatically generate an Internet address. You can change this behavior by using the -A option:

```
ommodu -o name/mailnode -A
```

This causes an Internet address to be generated for name. This replaces any existing Internet address for this user.

To change the Internet address of a user, you can use the following form of the ommodu command:

```
ommodu -o name/mailnode -n name/mailnode/internet-addr=internet_address
```

If you wish ommodu to generate Internet addresses automatically, without specifying -A or /internet-addr on the command line, set the configuration option INET_AUTOGEN_ON_MODIFY to TRUE (see “Configuration Options” on page 301).

Creating an Internet Address When You Add a PDL

By default, the omaddpdl command causes an Internet address attribute to be generated for the PDL you add, providing their mailnode is configured appropriately. You can change this behavior by using one of the following forms of the command:

- omaddpdl -l pdl_name/mailnode/internet-addr=internet_address

  internet_address is the full Internet address of the PDL, and will override the automatically-generated Internet address.

- omaddpdl -l pdl_name/mailnode -Q

  Use the -Q option to prevent an Internet address from being automatically generated for this PDL.

Creating an Internet Address When You Modify a PDL

By default, ommodpdl does not automatically generate an Internet address. You can change this behavior by using the -A option:

```
ommodpdl -o pdl_name/mailnode -A
```

This causes an Internet address to be generated for pdl_name. This replaces any existing Internet address for this PDL.

To change the Internet address of a PDL, you can use the following form of the ommodpdl command:

```
ommodpdl -o pdl_name/mailnode -l \ pdl_name/mailnode/internet-
addr=internet_address
```

If you wish ommodpdl to generate Internet addresses automatically, without specifying -A or /internet-addr on the command line, set the configuration option INET_AUTOGEN_ON_MODIFY to TRUE (see “Configuration Options” on page 301).

Removing an Internet Address from a User or PDL

To remove the Internet address from a user or PDL, you modify the user or PDL and specify a null Internet address. For example:

```
ommodu -o "Chris Wolf/ny,bigco,sales" -n "Chris Wolf/ny,bigco,sales"
```

This removes the user’s INTERNET-ADDR attribute, whether it was generated automatically or manually.

Aliases and Internet Addresses

When you create a user or PDL with an automatically generated Internet address, an Internet address is also generated for any aliases for that user or PDL.

For example:

```
omaddmn -m 'sales,us' -D 'sales.loc1.org.com'
omaddu -n 'John T. Smith/sales,us' -a 'Joe Brown'
```

John T. Smith/sales,us

will have an Internet address of:

John.T.Smith@sales.loc1.org.com.

The alias Joe Brown will have an Internet address of Joe.Brown@sales.loc1.org.com

If you manually specify an Internet address when you create a user or PDL, then an Internet address will still be generated automatically for any aliases. For example:

```
omaddmn -m 'sales,us' -D 'sales.loc1.org.com'
omaddu -n 'John T. Smith/sales,us/internet-addr= \
john.smith@hq.loc2.org.com' -a 'Joe Brown'
```

In this case, the alias Joe Brown will have the Internet address Joe.Brown@sales.loc1.org.com.

You can create an Internet address for an alias manually. For example:

```
omaddu -n "Chris Wolf/ny,bigco,sales/internet-addr= \
Chris.Wolf@sales.bigco.ny.com" \
-a "Joe Smith/ny,bigco,sales/internet-addr= \
jsmith@hq.bigco.ny.com" -a "Jim Smith/ny,bigco,sales"
```

This creates the user Chris Wolf and assigns two aliases. Joe Smith has the Internet address specified. If the mailnode ny,bigco,sales has an Internet domain, then Jim Smith will have an automatically generated Internet address.

Chapter 13

Local Delivery Service

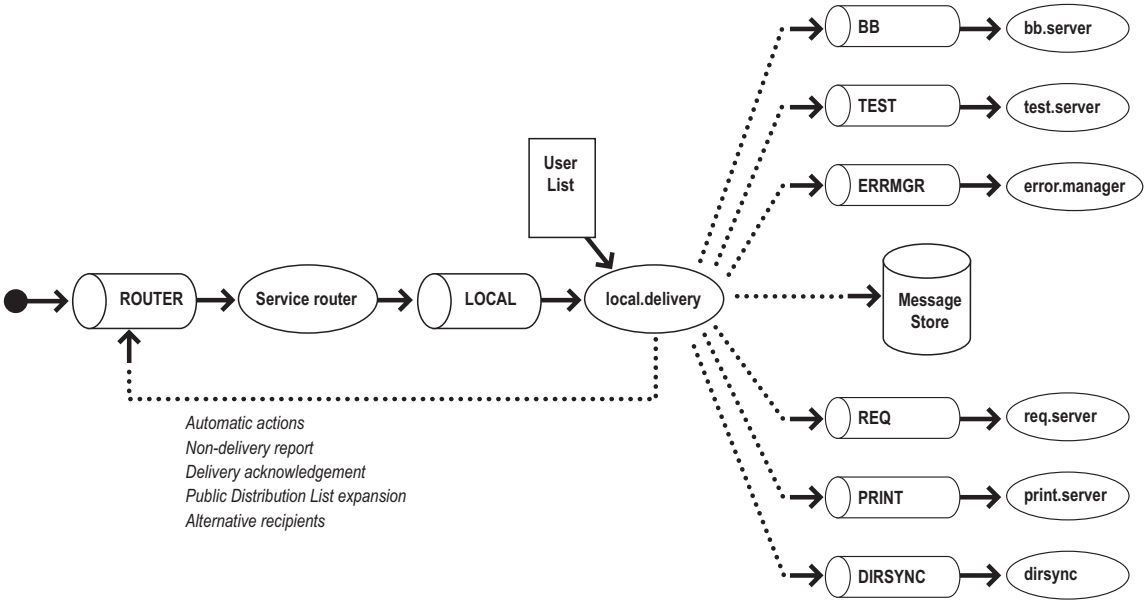
The Local Delivery Service handles local message delivery. Local messages may be addressed to a normal user, a server program, or a Public Distribution List.

However, when the option SR_LD_BYPASS_LSERV is set to TRUE in the General Configuration File (~/sys/general.cfg), the Local Delivery Service is (under certain conditions) bypassed.

This chapter includes the following information:

Local Delivery Service Overview

The following illustration shows how the Local Delivery Service handles local message delivery.



The following summarizes key Local Delivery Service information.

- Service name: local
- Process name: local.delivery
- Queue name: LOCAL

How the Local Delivery Service Handles Messages

The Service Router passes messages to the Local Delivery Service queue LOCAL. The Local Delivery Service reads the active recipient list of the message and determines if the latest delivery time for the message has expired.

If the latest delivery time has expired, the Local Delivery Service generates a Non-Delivery Report for the active recipients and returns the Non-Delivery Report to the originator.

If the latest delivery time has not expired or a latest delivery time is not specified, the Local Delivery Service continues to process the message. For each recipient O/R Address marked as "local", the Local Delivery Service refers to the User List, and follows this procedure:

- If the recipient cannot be found in the User List and there is no OSAR (Originator Specified Alternative Recipient), the Local Delivery Service generates a Non-Delivery Report (unless disabled by the originator). The message is then forwarded with the Non-Delivery Report to the originator and to the Error Manager (if one is configured).
- If an OSAR is specified, the Local Delivery Service removes the recipient O/R Address from the active recipient list and replaces it with the OSAR O/R Address. (The transaction file for the message records that an OSAR has been used.)

- If the recipient is a normal user, any automatic actions are performed once the message is attached to the user's In Tray. If appropriate, a Delivery Report is returned to the originator, and the message is delivered to the user's In Tray in the Message Store.
- If a recipient redirection is configured for the user, the Local Delivery Service recreates the message, removes the recipient O/R Address from the active recipient list, and replaces it with the new O/R Address for the recipient redirection. (The transaction file for the message records that a recipient redirection has been used.)
- If a recipient redirection is configured, the message is never actually delivered to the original recipient's In Tray before being redirected, unlike an autoforward.
- If the recipient is a server, the message is attached to the associated server queue, and if appropriate a Delivery Report is returned to the originator.
- If a message is addressed to a Public Distribution List, the active recipient list is expanded and a "recipient redirected" operation trace record is added to the transaction file for the message. This record prevents the Service Router from considering the message to have looped. If appropriate, a Delivery Report is also returned to the originator, unless the Delivery Report is addressed to a Public Distribution List, in which case it is discarded.
- If a Non-Delivery Report is addressed to a Public Distribution List, the Public Distribution List is not used, unless it is the designated Error Manager. The Non-Delivery Report is returned to the Error Manager instead of the members of the Public Distribution List.If the message is an X.400 probe, a Delivery Report is generated and returned to the originator.

When the Local Delivery Service has processed each recipient O/R Address marked local, the Local Delivery Service attaches the message for any expanded Public Distribution Lists, recipient redirections, OSARs, and delivery reports to the Service Router queue ROUTER.

Bypassing the Local Delivery Service

When the option SR_LD_BYPASS_LSERV is set to TRUE (the default setting) in the General Configuration File (~/sys/general.cfg), the Local Delivery Service can sometimes be bypassed, depending upon the nature of the message.

When set to TRUE the sequence of events that takes place is as follows: Firstly, a check is made, to find out whether the message in question is local. If it is, a further check is made to find out whether it is addressed to one of the following Scalix service queues:

- Bulletin Board Server
- Error Manager Server
- Request Server
- Print Server
- Directory Synchronization Server

If the message is addressed to one of the above queues, it is routed directly to the queue.

The main advantage of this process is that it considerably reduces the amount of time required for Directory synchronization, by minimizing traffic through the Local Delivery Service. In addition, it also increases the speed of other local traffic.

However, if this option is selected, ACLs limiting access to the Local Delivery Service can also be bypassed. See "Access Control Lists" on page 197.

For more information about SR_LD_BYPASS_LSERV, see “Configuration Options” on page 301.

Chapter 14

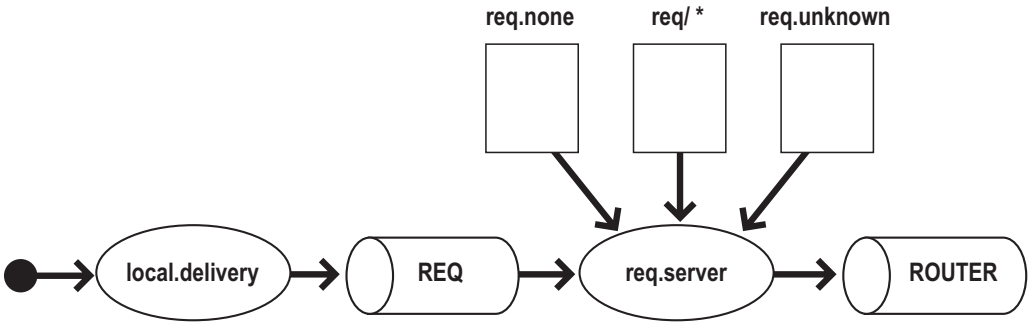
Request Server

The Request Server responds to messages addressed to it by running a script or program you have installed. The given name used in the address for the Request Server determines which script or program is run; the output of the script or program is then returned to the originator of the message. The Request Server is an entity that exists at all addresses in an Scalix network.

This chapter includes the following information:

Overview

The following illustration maps out the typical Request message routing.



1 Request messages are routed in the normal way by the Service Router.

   If the request message is for a local address it is passed to the Local Delivery Service by the Service Router.

2 The Local Delivery Service attaches messages addressed to the entity +REQ to the Request Server queue REQ.

3 The Request Server then attempts to execute the script or program /opt/scalix/req/request-name.

   - If /opt/scalix/req/request-name exists, it is executed. The standard output and standard error of the script or program is placed in a message and returned to the originator of the request message.
   - If /opt/scalix/req/request-name does not exist, the Request Server executes the script /opt/scalix/bin/req.unknown. This script, by default, returns a text message stating that the script or program could not be found.
   - If request-name is not supplied, the Request Server executes the script /opt/scalix/bin/req.none. This script, by default, returns a text message stating that a valid request-name must be used.

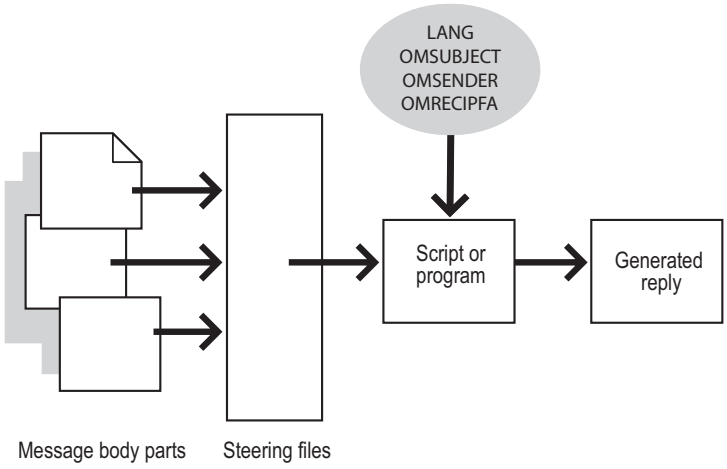
The following table lists key Request Server information:

Item	Name
Service name	request
Process name	req.server
Queue name	REQ
Given name	request-name
Surname	+REQ

Script and Program Requirements

Request server scripts and programs should operate as described in the following sections. (Examples of Request Server scripts can be found in the directory `/opt/scalix/examples/req`.)

Data Input and Output



The body parts of a request message can hold request information, which is additional to the subject. Body parts are converted according to the appropriate steering file. The script or program then reads standard input (`stdin`) for the request information. See “Steering Files” on page 177 for more information.

The script or program must read the environment variables listed in the following table for information about the request message:

Environment Variables	Description
LANG	The originator’s language. (Initially, this is held within the request message and mapped to `~/sys/LangMap` and Charmap to produce a suitable `LANG` value. The Request Server sets `LANG` to this value before the Request Server script runs.)
OMSENDER	The originator’s O/R Address in Scalix positional form.
OMSUBJECT	The subject of the message.
OMRECIPFA	The concatenated values of any Domain Defined Attributes (DDAs) placed in the address of the Request Server by the originator.

The script or program must write a reply to standard output (`stdout`).

Steering Files

You can send the Request Server additional data in the body part of the message. This data is used as additional input for the request script or program. The Request Server uses steer-

ing files to determine what file format and character set conversions to carry out before passing the data to the script or program. Note that, after body part conversion, the Request Server passes only the final remaining body part to the script or program, and ignores any others.

For example, you can send data in RTF format to the Request Server, and specify in the steering file that it is converted to plain text. The text is then passed to the Request Server as input data for the script or program.

The Request Server uses these steering files in order of preference:

- ~/scalix/req/*request-name*out.str

  request-name is the name of the request script or program.

- ~/scalix/sys/requestout.str

  The Request Server uses this steering file if the above steering file does not exist. You can edit this file to provide the required behavior. The unmodified version destroys any `winmail.dat` attachments (file type 1734) and passes other attachments unchanged.

Temporary Files

The script or program may create any temporary files it requires and must remove any temporary files it creates when it has finished or when it is stopped with a SIGTERM signal. Temporary files must have a file permission of 660.

Creation of Processes

The script or program can create child processes if necessary. When the script or program receives a SIGTERM signal, it must finish in a tidy way together with any child processes it has started.

Return Values

On successful completion, the script or program must return a value of zero. If an error occurs, the script or program must return a non-zero value (use values other than 1 because 1 is returned by the Bourne shell if there is a shell error).

Miscellaneous

The file name used for the script or program must contain lowercase letters and numbers only. The file name must not contain any character that could be interpreted as a metacharacter by the shell executing the request server script or program (characters such as *, _, or $). Also, where possible, avoid beginning names with the letters "om".

Configuring Requests

Requests are "configured" by placing a script or program in the directory `/opt/scalix/req`. Test the request by sending a message to the appropriate Request Server with the name of the script or program as the given name of the Request Server's O/R Address.

Note	Security Before placing a script or program in the /opt/scalix/req directory, examine it for possible security breaches. For example, do not use scripts that send back a copy of any requested file (password files in particular). Use Access Control Lists or passwords to control who uses a particular Request Server script or program. See "Restricting Access to the Request Server" on page 183 for more information.

Addressing the Request Server

The Request Server has a surname of +REQ and a given name of *request-name,* where *request-name* is the name of the request configured on the target Scalix Server.

For example, an Scalix Server, with a local address of `boston,factory,mis` configured on it, has a script in the `/opt/scalix/req` directory named audit. The script, when executed, copies the contents of the Audit Log file to its stdout and so into the return message.

To obtain a copy of the Audit Log File, address a message as follows:

```
TO: Audit +REQ / boston,factory,mis
```

Configuring Requests in the Directory

When sending a request message, the user must supply the full O/R Address, even if the message is for the Request Server on the local Scalix Server.

To simplify the addressing of often-used requests, the O/R Address of the request can be added to the local Directory using the command omaddent. This is the same as adding a user to the Directory. Where appropriate, the request can also be added to Directories on remote systems.

You can also use a Public Distribution List to give a more meaningful name to the request.

Example Uses for the Request Server

The range of different uses for the Request Server is limited only by your ability to write Unix shell scripts and programs. This section uses two case studies to show some of the common ways the Request Server can be used.

Case Study 1: Providing an Information Service

As the administrator of an Scalix system at Pinewood Inc., you have been asked to provide an automatic product information service. You decide to use the Request Server, and plan to keep a series of electronic product data sheets as text files on the Scalix system at the Boston factory.

To start with, you write a script that selects the product data sheet according to the subject of the request message. The script will read the correct text file, write it to stdout, and stop. You place the script in /opt/scalix/req and configure the request in the Directory. You call the script info.

This is an example of how the system is subsequently used.

1 Marion Brand, a member of the London sales staff, wants the latest information on Pinewood Inc's range of solid wood doors. She sends a message with the subject of doors to info +REQ / boston,factory,mis.

2 The Request Server on the Boston Scalix system starts the info script.

3 The script selects the doors text file and writes it to stdout.

4 The Request Server creates a message from the script's output and sends it back to the originator (Marion Brand).

5 Marion Brand receives the data sheet for solid wood doors.

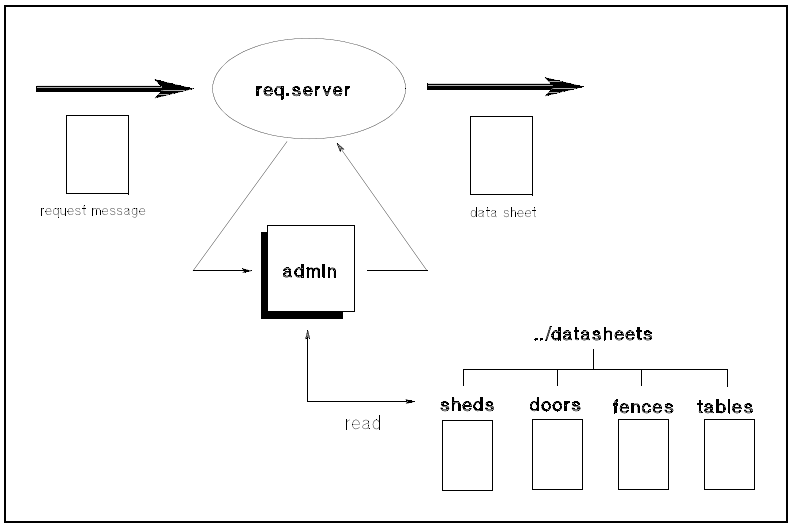
6 Kate de Ville, wants information on the range of garden sheds. She sends a message with the subject of sheds to info +REQ / boston,factory,mis.

7 The Request Server on the Boston Scalix system starts the info script.

8 The script selects the sheds text file and writes it to stdout.

The Request Server creates a message from the script's output and sends it back to the originator (Kate de Ville).

9 Kate de Ville receives the data sheet for garden sheds.



Case Study 1: Further Improvements

To improve the service and control who accesses it, you could use Access Control Lists on the Boston Request Server or the `info` script (see "Restricting Access to the Request Server" on page 183). Alternatively, you could change the script so it either checks the address of the originator (the environment variable OMSENDER contains the address), or you could change the script so it uses a password supplied in the body part of the message.

You could also consider other changes; for example, developing the information system to allow users to submit new data sheets to the information store using the Request Server. The submissions could be controlled by using the address of the submitter or passwords in the body part of the message.

Example Script for an Information Service

The following example script placed in the /opt/scalix/req directory of an Scalix Server will return information based on the subject of the message.

```
FILE="$OMSUBJECT"

 # document directory
INFOBASE=/usr/infobase
# check directory exists
cd $INFOBASE
if [ $? != 0 ] then
    echo "Information base not set up."
    exit 2
fi

# exclude names that are not base file names
case "$FILE" in
"") FILE="" ;;
*[$‘();|/]*) FILE="" ;; [.-]*) FILE="" ;;
*) if [ ! -r "$FILE" ]
           then
        FILE=""
     fi
     ;;
esac

# return requested file
if [ "X$FILE" != "X" ]
then
      cat "$FILE"
      exit 0
fi

# default message if no file or non-existent file is requested echo "This
information server will return a selected item based" echo "on the subject
of the message sent to it. The only items" echo "currently available are:"
echo ""
ls -C
exit 1
```

Case Study 2: Performing Remote Administration

As the administrator of the Scalix systems at Forester Holdings, you have to perform Scalix administration work at a number of different sites. To save traveling time, you have decided to use the Request Server to perform administration tasks remotely.

You write a Request Server script that runs the commands listed in the body part of the request message. The script will check your O/R Address against a list of O/R Addresses that are allowed to use it, produce a temporary script containing the commands (which it gives a

unique name, like a password), and write the confirmation text to stdout. You place the script in /opt/scalix/req and configure the request in the Directory. You call the script admin. You repeat this process for each Scalix system in the Forester Holdings network.

This is an example of how you could use the script.

1. You are working in the New York office and you want to delete George Sand from the Paris subsidiary's Scalix system. You send a message to admin +REQ / paris,sales,mis. In the last body part of the message you type /opt/scalix/bin/omdelu -n "George Sand".
2. The Request Server on the Paris Scalix system starts the admin script.
3. The script checks the originator O/R Address. (If the O/R Address is not one with sufficient authority, the script writes a message to stdout saying nothing has been done.)
4. The script reads stdin for the commands that have to be run. It produces a temporary script called zxdf1328g that runs the omdelu command.
5. The script writes a message with the temporary script name to stdout.
6. You receive a confirmation from the Request Server on the Paris Scalix system. The response includes a password of zxdf1328g that must used in the confirmation message.
7. You send message with subject of confirm to zxdf1328g +REQ / paris,sales,mis
8. The Request Server on the Paris Scalix system starts the zxdf1328g script.
9. The script checks the subject is confirm. If it is, the script runs the omdelu command.
10. The script writes a completed message to stdout.
11. You receive a message confirming that George Sand has been deleted from the Paris system.

Case Study 2: Further Improvements

You could simplify the script by removing the second temporary script stage. This is fine if you are sure will not need to change your mind, and you use passwords to make sure security is not breached.

Additionally, it is important to delete the temporary scripts when they have run. This is easily done using an entry in the crontab file. Each temporary script could be created so, when it has run, it adds a .done extension to its name to show it can be deleted. The crontab job could then remove all the .done scripts each night.

You could also add a crontab entry that looks for all temporary scripts that are older than, say, two days. If the crontab job finds an old script, it sends a message to the Error Manager with a copy of the script. The Error Manager, who is probably you, could then investigate what the script is meant to do, who should have run it, and what they were trying to do.

Restricting Access to the Request Server

To control access to the script or program using an Access Control List (ACL), make sure the file name of the script or program does not exceed 10 characters. The script or program must be executable for the user scalix. Scripts are executed by the Bourne shell.

For example, you have a script which can change user passwords (called chgpwd) and you only want to allow a user (Jadzia Dax) to be able to access it, do the following:

1 Add the ACL:

```
omaddacl -t req -l chgpwd
omshowacl -t req -l chgpwd
Scalix Administrators config execute
Local Users none
Default none
```

2 Remove the Admin capabilities:

```
ommodacln -t req -l chgpwd -g admin -c "-config -execute"
omshowacl -t req -l chgpwd
Scalix Administrators none
Local Users none
Default none
```

3 Add Jadzia Dax:

```
omaddacln -t req -l chgpwd -n "Jadzia Dax/telos,ds9" -c execute
omshowacl -t req -l chgpwd
Jadzia Dax /telos,ds9 execute

Scalix Administrators none
Local Users none
Default none
```

Chapter 15

IMAP4 and POP3 Servers

This chapter describes the IMAP4, LDAP, and POP3 servers for Scalix, and includes the following information:

About the IMAP4 Server

The Scalix IMAP4 Server enable clients to do the following:

- Access, list, read, and delete items from the inboxes and Public Folders
- Read parts of a message without downloading the entire message from the IMAP4 Server
- Keep a record of which messages have been read
- Update messages on the Server from a client

IMAP4 clients connect to Scalix through the IMAP4 Server daemon in.imap41d, which connects to Scalix using the User Agent Layer (UAL). The ual.remote process manages communications between the IMAP4 Server and Scalix. Scalix interprets the IMAP4 Server as a remote UAL client.

Note that the IMAP4 Server does not have to reside on the same machine as the Scalix Server to which it connects. For example, to reduce the load on the Scalix Server, you can run the IMAP4 Server on a separate machine. You specify the Scalix Server that the IMAP4 Server connects to by using the IMAP_MAILSTORE_HOST configuration option.

Scalix messages retrieved by IMAP4 clients are converted into MIME format. Steering files are used to configure the conversion of message body parts and the character set of textual body parts.

Steering files are held under ~/sys/. They are all text files, which you can edit using any standard text editor.

The following table lists the steering files that are used with the Scalix IMAP4 Server:

IMAP4 Server File	Description
brwmime.str	Controls the formatting of outgoing messages, including both body part and character set conversions that occur before the message is MIME-encoded. The format and function of this file is the same as `~/sys/mime-out.str`. If the **brwmime.str** file does not exist, the **mimeout.str** file is used.
mimeout.str	Outgoing body part conversions (for MIME. Used if **brwmime.str** is not present).
unixout.rules	Outgoing name conversions.
unixmap.out	Outgoing address mappings.
mime.types	Controls the mapping of content-types (and their subtypes) to body parts, including the following MIME content-types: Text/enriched Image *Video* Application The file codes specified in the **mime.types** determine the file mapping performed. If a mapping is not present in the **mime.types** file, the body part is marked as binary (file code 0) except for content-types of "*multipart*" and "*message*", which are treated as special cases. The MIME content type "*text/plain*" is mapped to a textual body part (file code 1167) of the same character set. Further mappings between file codes and content-types can be specified in this file. The format of the **mime.types** file is documented in the header of the file.
mime.cs	Maps character set names used in MIME to those used in Scalix. Each mapping you define in the **mime.cs** file must be on a separate line, and in the following format: Scalix_Character_Set_Name [TAB] MIME_Character_Set_Name By default, text goes out in the ISO_8859_1 character set. This is encoded as 7-bit if there are only ASCII characters, or quoted-printable if it contains 8-bit data. Other (binary) attachments are base64-encoded.

IMAP4 Name and Address Mapping

To use Scalix Directory-based address mapping, set the following option in the general configuration file (`~/sys/`**general.cfg**):

```
BRW_NAME_MAPPING=TRUE
```

This performs Directory lookups on the Scalix Directory specified by the UX_NAME_MAPPING_DIR option.

Configuring IMAP4

The following information describes how to configure an IMAP4 server and client.

Configuring the IMAP4 Server

After installing or updating Scalix, follow these steps to configure the IMAP4 Server:

1 Make sure the following line exists in `/etc/services`:

```
imap4 143/tcp
```

2 Refresh the `inetd` daemon.

```
inetd -c
```

Refreshing the daemon ensures it reads the updated files.

The IMAP4 interface will function when the remote UAL is enabled.

Configuring IMAP4 Clients

For each client, you must configure the host name of the IMAP4 Server and the user name that will be used to sign on to Scalix.

Some clients do not operate if the user name contains some special characters such as a space, forward slash (/), or comma (,). Some clients can not allow you to specify a user name in this form.

To enable IMAP4 client users to sign on with a name such as `John Doe /ny,hq,` use Scalix aliases. Scalix aliases for a user can either be specified when the user is created (using `omaddu`) or added later (using `ommodu`) with the `-a` option. See the relevant online manual entries for more details of these commands.

To enable login using aliases, set the following options in the `~/sys/`**general.cfg** file:

```
UAL_SIGNON_ALIAS_CONFIG

UAL_SIGNON_ALIAS

UAL_USE_SIGNON_ALIAS
```

For details of these options and other options that allow you to modify the behavior of IMAP4 clients, see “Configuration Options” on page 301.

Note	To ensure that IMAP4 folders with names containing multibyte characters are correctly displayed by the client, you must configure the relevant Scalix users with the correct language. Do this by using the -l option with the omaddu or ommodu commands.

About the POP3 Server

The Scalix POP3 Server enables POP3 clients to list, read, and delete items from the Inbox area of the Scalix Message Store. The Scalix POP3 Server does not provide access to any other areas of the Message Store, such as the Public Folder area. For access to this and other areas, see “About the POP3 Server” on page 187.

POP3 clients connect to Scalix through the POP3 Server process.

POP3 Message Body Part Conversions and File Types

Scalix messages retrieved by POP3 clients are converted into MIME format. Steering files are used to configure the conversion of message body parts and the character set of textual body parts.

Steering files are in the ~/sys/ directory. They are text files that you can edit using any standard text editor.

The following steering files are used with the Scalix POP3 Server:

File Name	Description
brwmime.str	Outgoing body part conversions (default). Controls the formatting of outgoing messages, including both body part and character set conversions that occur before the message is MIME-encoded. The format and function of this file is the same as ~/sys/mimeout.str. If the brwmime.str file does not exist, the mimeout.str file is used.
mimeout.str	Outgoing body part conversions (for MIME - used if brwmime.str is not present).
unixout.rules	Outgoing name conversions.
unixmap.out	Outgoing address mappings.
mime.types	File code to MIME content-type mappings. Controls the mapping of content-types (and their subtypes) to body parts, including the following MIME content-types: Text/enriched Image Video Application The file codes specified in the mime.types determine the file mapping performed. If a mapping is not present in the mime.types file, the body part is marked as binary (file code 0) except for content-types of "*multipart*" and "*message*", which are treated as special cases. The MIME content type "*text/plain*" is mapped to a textual body part (file code 1167) of the same character set. Further mappings between file codes and content-types can be specified in this file. The format of the mime.types file is documented in the header of the file.
mime.cs	Maps character set names used in MIME to those used in Scalix. Each mapping you define in the mime.cs file must be on a separate line, and in the following format: Scalix_Character_Set_Name [TAB] MIME_Character_Set_Name By default, text goes out in the ISO_8859_1 character set. This is encoded as 7-bit if there are only ASCII characters, or quoted-printable if it contains 8-bit data. Other (binary) attachments are base64-encoded.

POP3 Name and Address Mapping

To use Scalix Directory-based address mapping, set the following option in the general configuration file (~/sys/**general.cfg**):

```
BRW_NAME_MAPPING=TRUE
```

This performs Directory lookups on the Scalix Directory specified by the UX_NAME_MAPPING_DIR option. For details of these and other POP3 options, see “Configuration Options” on page 301.

Configuring POP3

The following information describes how to configure an IMAP4 server and client.

Configuring the POP3 Server

After installing Scalix, follow these steps to configure the POP3 Server:

1 Make sure the following line exists in /etc/services:

```
pop3 110/tcp
```

2 Refresh the inetd daemon. Enter:

```
inetd -c
```

Refreshing the daemon ensures it reads the updated files.

Configuring the POP3 Client

For each client, you need to configure the host name of the POP3 Server and the user name that will be used to log in to Scalix.

Some clients do not operate if the user name contains some special characters such as a space, forward slash (/), or comma (,). Some clients can not allow you to specify a user name in this form.

Use Scalix aliases to enable POP3 client users to sign on with a name such as John Doe / ny,hq,

Scalix aliases for a user can either be specified when the user is created (using omaddu) or added later (using ommodu) with the -a option. See the relevant online manual entries for more details of these commands.

To enable login using aliases, set the following options in the ~/sys/**general.cfg** file:

```
UAL_SIGNON_ALIAS_CONFIG
UAL_SIGNON_ALIAS
UAL_USE_SIGNON_ALIAS
```

For details of these options, see “Configuration Options” on page 301.

Chapter 16

The LDAP Server

The LDAP Server is based on a global directory model named the Lightweight Directory Access Protocol (LDAP). LDAP is an Internet directory service protocol that runs over TCP/IP. This chapter includes the following information about the LDAP server:

About the LDAP Server

The LDAP Server is a Scalix daemon process that provides an interface to enable LDAP clients to store and retrieve data from a Scalix Directory without having any information about the operation of Scalix.

The LDAP Directory service is based on a client-server model. The LDAP Server provides LDAP clients access to shared Scalix Directories that do not have an associated password.

Scalix automatically enables search-only LDAP support. Consequently, there is minimal configuration required to enable LDAP client directory searches. The LDAP Server process (omslapd) starts when Scalix starts and runs until Scalix is shut down.

In LDAP, Directory entries are arranged in a hierarchical tree-like structure that reflects political, geographic, and/or organizational boundaries. This structure is called a Directory Information Tree (DIT). The LDAP Directory is a DIT comprising one Scalix Directory containing structural information and one or more additional Scalix Directories containing user and entity information. Using this structure, the LDAP Server provides a hierarchical view of a Scalix Directory, enabling LDAP clients to access Directory entries.

An entry is referenced by its Distinguished Name (DN), also known as a Directory Distinguished Name (DDN), which is an unambiguous identifier for that entry. The DN is constructed from a Relative Distinguished Name (RDN).

The name of a required LDAP attribute is not necessarily the same as the attribute stored in the Scalix Directory. Therefore, a configuration file is used to map attributes between LDAP and Scalix attribute type names. For names which have no corresponding entry in the other Directory, parsing and synthesis methods provide a mechanism for mapping between explicit

LDAP attribute types and attribute types that are only implicitly defined in the Scalix Directory.

The behavior of the LDAP Server is controlled by a number of configuration files.

How the LDAP Server Works

The LDAP Directory service is based on a client-server model. One or more LDAP directories contain the data making up the LDAP Directory Information Tree. In the Scalix LDAP Server, the DIT is comprised of one Scalix Directory containing structural information (the *Scalix DIT Directory*) and one or more additional Scalix Directories containing entries identifying users or entities. The client-server relationship works as follows:

1. An LDAP client connects to an LDAP Server.
2. The LDAP client issues a bind request to the LDAP Server.
3. The LDAP client issues a search or update request to the LDAP Server.
4. The server either returns the results or refers the client to another LDAP Server where it can get more information on the requested data.
5. The LDAP client unbinds from the LDAP Server.
6. The LDAP client disconnects from the LDAP Server.

To access a Scalix Directory through the LDAP Server, the LDAP client binds to the LDAP Server. The user specifies one of the following types of bind as part of the search or update request:

- anonymous bind: Read-only access to the directory entries is usually provided without any user authentication. This is the usual type of bind performed.
- simple bind: The Scalix user's ID and password are supplied in the bind request. The ACL for the relevant directory is then checked to determine the type of access that the user has. For example, Scalix administrators or Directory owners can add, delete, or modify Directory entries.

Following a successful bind, the LDAP client submits its request to the LDAP Server. For an update request, the LDAP Server must first ensure that it owns the parent of the specified entry; while for a search request it only needs to own a copy of the requested entries.

LDAP Server Configuration Files

The LDAP Server is controlled by configuration files which allow you to customize the operation of the LDAP Server for your installation of Scalix. Modify the following configuration files listed in this table, according to your requirements:

File Name	Description
`ldap.attribs`	LDAP attribute mapping file. This file defines the mapping between LDAP attribute names and Scalix internal attribute names.
`slapd.conf`	LDAP Server configuration file. This file sets options that control the runtime behavior of the LDAP Server.

File Name	Description
dit.cfg	DIT configuration file. This file specifies the name of the Scalix DIT Directory and the default DN suffix used by some of the Scalix Directory commands.

Starting and Stopping the LDAP Server

The LDAP Server process (omslapd) starts when Scalix starts and runs until Scalix is shut down. If required, you can stop the LDAP Server by entering:

```
omoff -d delay -a slapd
```

delay indicates the time in seconds to wait before stopping the daemon.

To start the LDAP daemon process, enter:

```
omon -a slapd
```

LDAP Directories

The LDAP Directory service model is based on entries. An entry is a collection of attributes that has a name, called a Distinguished Name (DN). The DN is used to refer to the entry unambiguously. Each of the entry's attributes has a type and one or more values. The types are strings, like "cn" for common name, or "mail" for e-mail address. The values depend on what type of attribute it is. For example, a mail attribute might contain the value John-Doe@Acme.co.uk.

In LDAP, Directory entries are arranged in a hierarchical tree-like structure that reflects political, geographic, and/or organizational boundaries. Entries representing countries appear at the top of the tree. Below them are entries representing states or national organizations. Below them might be entries representing people, organizational units, and entities (such as printers or documents).

While LDAP has a hierarchical structure like a tree, the Scalix Directory has a flat structure like a telephone directory. The Scalix Directory is made up of a series of entries identifying a user (or entity) by attributes. Attributes include O/R Address attributes, personal and employment related attributes, and e-mail address attributes among others. Traditionally, the Scalix Directory is a single, flat database. Entries are not grouped into any hierarchal structure.

Searching for Entries in LDAP and Scalix Directories

LDAP defines operations for interrogating and updating the Directory. However, LDAP is used primarily to search for information in a Directory. The LDAP search operation allows certain portions of the Directory to be searched for entries that match criteria specified by a search filter. Information can be requested from each entry that matches the criteria.

For example, you might want to search the entire Directory sub-tree below the O=Acme Holdings entry for people with the name "John Doe", retrieving the e-mail address of each entry found. Or you might want to search the entries directly below the C=USA entry for organizations that have the string "Acme" in their name and have a telephone number.

An entry in a Scalix Directory is retrieved using a search filter, which is compared with the entries in the Directory. All entries in the Directory that match the filter are returned. For example, you could search the Directory for entries containing the Surname “Doe”.

Many attributes in a Scalix Directory are keyed (that is, indexed) to facilitate faster retrieval of frequently searched attributes. In general, full sequential searches of the directory are not performed, so searching for non-keyed attributes will fail to find a match. However, you can change this behavior by setting the general configuration option LDAP_SEQUENTIAL_SEARCH.

The difference in the LDAP and Scalix Directory structures is bridged by the LDAP Server, which enables users or LDAP clients to access the Scalix Directory through the Internet LDAP protocol. The LDAP Server allows hierarchical structures to be superimposed on existing Scalix Directories. This enables the Scalix Directory to be seen as a Directory Information Tree.

Note	If you wish to use LDAP only to search the Scalix directories, but not modify or add entries, no special configuration is required. Note, in particular, that you do not need to set up a DIT directory, although this will result in a flat LDAP directory structure.

LDAP and Scalix Attribute Type Mappings

The attribute type names of the Scalix Directory are, in general, different to those used by the LDAP protocol. This difference requires mapping between Scalix internal attribute names and LDAP attribute names.

In most cases there is a simple one-to-one mapping between LDAP attribute names and Scalix internal attribute names. However, some attributes used by LDAP (for example, mhsORAddress and commonName) might not be explicitly present in the Scalix Directory. When returning such values to the LDAP client, a complex attribute value is synthesized from a number of Scalix attributes.

Correspondingly, complex LDAP attribute values are parsed to construct appropriate filters or attribute requests before performing requests on the Scalix Directory.

This mapping is controlled by the LDAP attribute mapping file (~/sys/ldap.attribs), which contains the mapping between Scalix internal attribute names and LDAP attribute names. This mapping file is used for both LDAP to Scalix Attribute Type Mapping and Scalix to LDAP Attribute Type Mapping.

LDAP to Scalix Attribute Type Mapping

LDAP named attributes must be mapped to corresponding Scalix attributes when setting up the following:

- Search filters
- Lists of required attribute values.

This can be a simple one-to-one mapping or might require a more complicated parsing method. A single pass through the mapping function with a given LDAP attribute name and, optionally, a value produces a list of corresponding Scalix internal attributes and values.

If no parsing method type is specified in the mapping file, the corresponding Scalix attribute name is found. The mapping file expects corresponding attributes to have equivalent syntax.

The following table lists special LDAP attribute types and how they are parsed.

Attribute Type	Description
commonName,omAddress	These attributes can be parsed as Scalix addresses to obtain subordinate attributes when constructing a Scalix search filter from an LDAP search filter.
mhsORAddress	This attribute is parsed as an X.400 address to obtain subordinate attributes when constructing a Scalix search filter from an LDAP search filter.
objectClass	When constructing the search filter corresponding to an `objectClass` and the specified `objectClass` is that of a distribution list, the search filter is extended to not only match that `objectClass` but also to match any entry that has a Surname attribute and one or more distribution list members. Correspondingly, if the `objectClass` specifies a person, then the search filter is extended to match any entry that has a `Surname` attribute and *no* distribution list members. All entries in the Scalix DIT Directory have an explicit `objectClass` attribute, though these attributes do not usually correspond to people or distribution lists. Entries in other Scalix Directories typically do not have explicit object classes.

Scalix to LDAP Attribute Type Mapping

A search of the Scalix Directory returns a number of Scalix attributes and their values. These must be mapped to corresponding LDAP attribute names and values. This can be a simple one-to-one mapping process or a more complicated synthesizing process. All the LDAP requested attributes are searched for in the result and mapped with their values where possible. If all attributes have been requested, names and values are returned only when they can be found.

In a simple mapping, the internal Scalix attribute name is replaced with the principal LDAP attribute name, and its values are returned with any required changes to syntax representation. The mapping software does not handle attributes with compound syntax, such as DL-MEMBERS.

For each attribute type that has a synthesis method (see <Xref_Color>“LDAP and Scalix Attribute Type Mappings”), the corresponding attribute value is constructed from Scalix composite attributes as described in this table.

Attribute	Description
commonName	The LDAP common name attribute value is generated from the Scalix surname and given name attributes provided that the Scalix Directory entry does not contain a Common Name (`2.5.4.3`) attribute value. If `TELETEX-STRING` versions of surname and given name exist (Scalix attributes `51` and `52`, respectively), these are used rather than the `PRINTABLE-STRING` versions (Scalix attributes `1` and `2`, respectively).
inetAddress	If no `INTERNET-ADDR` attribute exists in the Scalix Directory, it is constructed from the Scalix Address attribute values in the Directory entry.

Attribute	Description
mhsORAddress	The LDAP attribute is constructed from those Directory attributes that form part of an X.400 address.
objectClass	If no explicit LDAP `objectClass` attribute exists in the Scalix Directory, it is mapped either to the `ScalixPerson` or the `ScalixDistributionList` attribute as required.
omAddress	This attribute is constructed from the Scalix Address attribute values in the Directory entry.

LDAP Commands

The following table lists and describes LDAP commands:

Command	Description
omldapadd	Add one or more entries to an LDAP Directory
omldapdelete	Delete one or more entries from an LDAP Directory
omldapmodify	Modify an LDAP Directory entry
omldapmoddn	Modify the DN of an LDAP entry
omldapsearch	Search an LDAP Directory

Chapter 17

Access Control Lists

This chapter describes Access Control Lists (ACLs) which control user access to Scalix resources. The following topics are detailed:

About ACLs

An ACL is associated with a resource, such as a service or Directory. The ACL lists users and the permissions they have for that resource. Users are listed by standard groupings; local administrators (admin), local users (local), and everyone/default (default). Additionally, specific users can be listed by their O/R Address or grouped using "wildcards" in place of specific address attributes.

The omaddacl and omdelacl commands create and delete ACLs for all resource types except Public Folders and Directories. Public Folder and Directory ACLs are automatically created and deleted when a Public Folder or Directory is created or deleted. The omshowacl command lists the permissions of users associated with an ACL.

The commands omaddacln, ommodacln, omdelacln, and omshowacl, add, modify, delete, and display the permissions of specific users or groups, respectively. Also, omchckacln allows you to verify the permissions of a user in an Access Control List.

Permissions can be removed using the ommodacln command and a dash (-) in front of the permission you want to remove.

The ACL configuration file acl.cfg is in the ~/sys directory.

The ACLs themselves are in directories under ~/acl.

Using ACLs

The following information describes how to use commands to create and manage ACLs.

Creating an ACL

You can create an ACL using the command:

```
omaddacl -t type -l name
```

Where type is the type of resource.

An ACL is identified by its "type" and "name". The following table lists the resource types that are available:

Resource	Type	Abbreviations	
Services	service	svc	s
Request Server scripts	request server	req	r
Public Folders	bulletin	bb	b
Directories	directory	dir	d

The resource types are defined in the ~/sys/acl.cfg file.

Every resource has configuration permission; that is, the permission to modify the ACL. This permission is always given to the standard group "local administrators" (and the root user) when the ACL is created. It can be changed later.

The name of an ACL depends on its resource type. The following table lists how ACL names are determined:

Resource	Name determined by
Services	the queue name of the service.
Request Server scripts	the file name of the request as listed in the `/opt/scalix/req` directory.
Public Folders	the temporary or absolute reference number of the Public Folder as listed by the `omlistbbs` command. (Use `0` for the Public Folder area itself.)
Directories	the name of the Directory as listed by the `omlistdirs` command.

Each entry in an ACL refers either to a standard group or to an O/R Address pattern.

Adding a User to a Service ACL

You can add a user to a service Access Control List (ACL) using the command:

```
omaddacln -t service -l queue_name
```

Adding a User to a Directory ACL

You can add a user to a Directory ACL using the command:

```
omaddacln -t directory -l Dir_name
```

Adding a User to a Public Folder ACL

You can add a user to a Public Folder ACL using the command:

```
omaddacln -t bulletin -l BB_ref -n <user> -c <caps>
```

The following table lists the levels of access that can be given to users or groups of users.

Permission	Description
see	Users can see the subject of the Public Folder. They cannot open the Public Folder or see the individual items attached to it.
read	Users can open the Public Folder and read the items attached to it.
attach	Users can attach items to the Public Folder.
delete	Users can delete any items from the Public Folder.
config	Users have full access to the Public Folder itself: they can delete it or rename it, for example. They can also add, modify, and delete ACL entries for the Public Folder. However, users cannot add items to the Public Folder or delete items from it.

Note that, if the attached item is itself a Public Folder, access to it is determined by its own ACL, and not the ACL of its parent Public Folder. For example, a user can have delete access to Public Folder A, giving them the permission to delete attached items. However, if Public Folder B is attached to Public Folder A, this user does not automatically have the permission to delete it. Such a user would require config access on Public Folder B in order to have this permission.

You can also modify an ACL by deleting the ACL file, using the omdelacl command. This causes the ACL to change from explicit to implicit, and the entries change from those in the ACL file to those of the parent Public Folder's ACL. If you delete the ACL file of a top-level Public Folder, its ACL changes to the default.

Default ACLs

The following table lists the default ACL for a top-level Public Folder.

User	Permission
Scalix Administrators	Config, Read, See, Delete, Attach
Other Users	Read, See, Attach

You can change the default level of access that "other users" are granted by setting the general configuration option UAL_FLDR_ACL_DEFAULT. When you create a nested Public Folder, it has a default ACL that is copied from its parent Public Folder.

Implicit and Explicit ACLs

A Public Folder has an "explicit ACL" when it has an associated ACL file. An explicit ACL is determined from the entries in the ACL file.

The ACL file is located in ~scalix/acl/bb.

A Public Folder has an "implicit ACL" when it has no associated ACL file. An implicit ACL is determined from the ACL of its immediate parent. In the case of a top-level Public Folder with an implicit ACL, its ACL is the default ACL as described in the section "Default ACLs".

When you first create a Public Folder, it has an implicit ACL. You create an associated ACL file for it when you modify its ACL for the first time.

You modify a Public Folder ACL by using the omaddacln, ommodacln, or omdelacln commands. In addition, some clients enable you to modify Public Folder ACLs.

When you modify the ACL, you convert an implicit ACL into an explicit ACL.

To see which Public Folders have explicit ACLs, issue the omlistbbs command with the -S option. Those Public Folders that have associated ACL files are indicated with a + symbol.

ACL Inheritance

An implicit ACL is inherited from the ACL of its parent Public Folder. For example, a Public Folder called Sales has a nested Public Folder called North. If North does not have an associated ACL file, its ACL will be the same as the ACL of Sales. If you change the ACL of Sales, the ACL of North will be updated automatically to reflect this change.

If North also has a nested Public Folder, Northwest, without an ACL file, then Northwest will also have an ACL that is the same as that of North, and of Sales.

As soon as you modify an implicit ACL, you make it explicit, and it no longer inherits its settings from its parent.

When you delete a Public Folder's ACL file, its ACL changes from explicit to implicit, and the ACL settings change from those in the ACL file to those of its parent.

You can ensure that all Public Folders inherit their ACLs from a specified parent Public Folder by deleting all ACL files below this parent in the hierarchy.

You can delete Public Folder ACL files using the omdelacl -t bb command. Use the -R option to delete the ACL together with the ACLs of all Public Folders below it in the hierarchy. See the man page for omdelacl for more information.

ACL Address Patterns

The following rules apply when specifying O/R Address patterns in ACLs:

- The O/R Address attributes by which a user is identified in an ACL are restricted to the mnemonic address form and are hierarchically ordered:

a. Country Name
b. Administration Domain Name
c. Private Domain Name
d. Organization Name
e. Organization Unit Name 1
f. Organization Unit Name 2
g. Organization Unit Name 3
h. Organization Unit Name 4
i. Personal Name

- An attribute value must be fully specified, partly represented with a wildcard, wholly represented with a wildcard, or left blank.

  The wildcard character is an asterisk (*). The wildcard represents zero or more characters.

To partly represent an attribute value with a wildcard, place the wildcard character at the end of the partial value. For example:

- `O=Pinewood` is *fully specified*
- `O=Pine*` is *partly represented with a wildcard*
- `O=*` is *wholly represented with a wildcard*

If an attribute value contains a wildcard, either wholly represent all less significant attributes with a wildcard or leave them blank.

If an Organizational Unit Name is left blank, leave all less significant Organizational Unit Names blank.

The Organization Name, Organizational Unit Name, and Personal Name attributes are specified in either printable strings or teletex strings, or both. If both forms are specified, and one form of the attribute value is represented by a wildcard, then the other form must be represented also with a wildcard to the same extent.

Matching Addresses to O/R Address Patterns in ACLs

The following rules are used when matching the O/R Address of a user to an O/R Address pattern in an ACL entry:

- Match characters regardless of whether they are uppercase or lowercase.
- Ignore address attributes that are not used by the mnemonic address form.
- If an address pattern specifies both printable and teletex strings for an attribute, and the address being matched contains one form only, then the other form in the address pattern is ignored.

Match each attribute:

- A specified attribute matches if each character compares "one for one".
- An attribute partly represented with a wildcard matches if each character in the specified part of the attribute compares one for one.

- A blank attribute matches if the attribute is also blank in the address of the user.
- An attribute wholly represented by a wildcard matches anything.
- If the address of the user matches an address pattern, the user is granted the capabilities specified for that address pattern.

Examples of O/R Address Patterns in ACLs

The O/R Address for a user named "John Doe" is:

```
G=John/S=Doe/CN=John Doe
OU1=paris/OU2=sales/OU3=mis
O=pinewood/P=forester/A=atlas/C=fr
```

It matches the following address patterns:

```
*/CN=*/OU1=paris/OU2=*/OU3=*/O=pinewood/P=forester/A=atlas/C=fr
*/CN=*/OU1=*/OU2=*/OU3=*/OU4=*/O=*/P=forester/A=atlas/C=fr
*/CN=*/OU1=*/OU2=*/OU3=*/OU4=*/O=*/P=for*/A=atlas/C=fr
G=John/S=Doe/CN=John Doe/OU1=*/OU2=*/OU3=*/OU4=*/O=*/P=*/A=*/C=*
```

The address for "John Doe" does not match the following valid address patterns:

```
*/CN=*/OU1=paris/OU2=*/O=pinewood/P=forester/A=atlas/C=fr
*/CN=*/OU1=paris/OU2=sales/OU3=mis/P=forester/A=atlas/C=fr
G-TX=John/S-TX=Doe/CN-TX=*/OU1=*/OU2=*/OU3=*/OU4=*/O=*/P=*/A=*/C=*
*/OU1=*/OU2=*/OU3=*/OU4=*/O=*/P=for*/A=atlas/C=fr
```

The first address pattern does not specify an OU3. The address for John Doe does have an OU3 equal to mis. Similarly, the Organization Name in the second address pattern is blank. In the third address pattern, only the teletex string version of the Personal Name is specified rather than the printable string version. In the fourth address pattern, the Common Name is blank.

The following address patterns are not valid:

```
*/CN=*/OU1=paris/OU2=sales/OU3=mis/O=*/P=forester/A=atlas/C=fr
*/CN=*/OU1=paris/OU3=mis/O=pinewood/P=forester/A=atlas/C=fr
*/CN=*/OU1=paris/OU2=sales/OU3=mis/O=*pinewood/P=forester/A=atlas/C=fr
```

In the first address pattern, the Organization Name is represented with a wildcard. Because of the lesser significance of the Organization Unit Names, these should also be represented with wildcards. In the second address pattern, OU2 is blank. Therefore, OU3 must also be blank. In the third address pattern, the wildcard character is incorrectly positioned in the Organization Name. The wildcard character must be at the end of the partial value.

Combining Users and Permissions

If a specific user is a member of more than one group, the permissions given to each group are combined for that user. The following example shows a Directory ACL:

```
John Doe/ny,hq,mis
                                update
*/ny,*
                              modifyself
```

```
*/*,*,*,*/*/*/*/*
                                read
Local Administrators
                               config
Local Users
                                none
Default
                                none
```

The omshowacl command outputs data in positional format.

The local user John Doe is explicitly given update permission. The user is also part of the group */*,*,*,*/*/*/*/* and so has read permission.

If John Doe is an Administrator for this system, then John Doe also has config permission.

Note that John Doe/ny,hq,mis does not match */ny,* because a wildcard has not been specified for the third Organizational Unit Name.

When a user attempts to use a resource, Scalix searches the ACL entries of the resource as follows:

1. The group entries in the following order:
   - Default
   - Local users
   - Local administrators
2. The address entries for an address pattern match.

This search order is used for all ACL entries. Because the group entries are checked in ascending order, a local administrator has the combined permissions of a default user and a local user, and a local user also has the permissions of a default user. If the group entries give sufficient permissions, the address entries are not searched. If Scalix does not find sufficient permissions within the group entries, it starts checking the address entries. If Scalix does not find an address match in the address entries, permission is denied.

ACL Commands

The following table lists commands associated with Access Control Lists.

Command	Description
omaddacl	Add an Access Control List
omaddacln	Add permissions for users to an Access Control List
omchkacln	Check permissions of a user in an Access Control List

Command	Description
omdelacl	Delete an Access Control List
omdelacln	Delete permissions for users from an Access Control List
ommodacln	Modify permissions for users in an Access Control List
omshowacl	Show the contents of an Access Control List

Chapter 18

Virus and Spam Protection

This chapter describes security measures you can implement across the Scalix messaging system. This chapter includes the following information:

Security Overview

The following information describes the security measures that you can use to protect the Scalix messaging environment.

Program and Data Security

Scalix data is owned by a special Linux user named scalix in the group named scalix. This user and group are created when you install Scalix. To prevent unauthorized access to Scalix data, make sure that other users do not belong to the group scalix.

Note	Most Scalix programs are owned by the Linux user scalix. Some programs are also owned and operated by the Linux root user.

The omcheck command enables you to verify that ownerships and permissions are set correctly for Scalix system files and directories.

Users access their data by being registered as Scalix users, and you can specify a password for each user. If you specify a password for a user, the user must enter the password before accessing their Scalix data.

Administration Capability

Scalix is only as secure as your operating system. Anyone with "root" permissions has unlimited access to Scalix resources. The ability to add, delete, or modify users or modify the Scalix system is restricted to users who are administrators or have root permissions.

Restricted User Access

You can control access to the various Scalix services and Directories using Access Control Lists. See “Access Control Lists” on page 197 for more information.

Message Security

There are instances when a Scalix Administrator can read messages addressed to other people, for example, as part of a Non-Delivery Report or when using some Scalix diagnostic tools. Messages marked as "Personal", "Private", or "Company Confidential" cannot be read by a Scalix Administrator.

Monitoring Usage

The Audit Log can record user activity. The Audit Log can identify unusual usage patterns or provide evidence of when individuals were using the Scalix system.

Virus and Spam Protection

The Service Router interfaces with virus-scanning applications from Trend Micro and McAfee. When you activate virus scanning, the Service Router scans all Scalix message attachments. Depending on how you configure virus scanning, the Scalix Server can attempt to repair infected files, return infected messages to the sender, or discard the message. See “Anti-virus Overview” on page 206 for more information.

Scalix also allows you to configure anti-spam measures on the SMTP Relay to prevent abuse of the Scalix system by external entities. See “SMTP Authentication and Anti-spam Protection” on page 211 for more information.

Client Security

Outlook E-mail Security parameters provides protection against software viruses that users might receive in their Inbox as an attachment file. To configure security for the MS Outlook client, see “Microsoft Outlook Security Model” on page 225.

Anti-virus Overview

The Scalix Virus Protection Framework enables you to scan for viruses attached to messages on a Scalix Server. If a virus is detected in a message, you can configure Scalix to discard the message or repair the infected attachment.

However, Scalix does not perform the actual virus scanning tasks. Instead, Scalix operates with the following third-party anti-virus applications (which support a command-line interface):

- McAfee VirusScan for Linux
- Trend Micro InterScan VirusWall
- Clam Anti-Virus

The Scalix Virus Protection Framework extends the Scalix message delivery rules by including additional rulesets and a "mapper" script. This script, omvscan.map, is always enabled and associate calls from the Scalix Service Router process to the appropriate third-party virus scanning command. omvscan.map is in the ~/rules/ directory.

Note	If you are implementing Clam Anti-Virus, you must edit the /etc/group file and change the scalix:n:xx to scalix:n:xx:clamav. Making this modification ensures that the clamAV does not encounter problems (with permissions) while scanning files on the Scalix Server.

Message Delivery Rules

A message delivery rule causes the Service Router to test the value of a message attribute, and to carry out specific actions based on the result of the test. For example, you can create a rule to test if the Priority of a message is Low and, if true, defer the delivery of the message until after normal office hours.

Rules are contained in rulesets, each of which can contain one or more rules. Therefore, a ruleset can test for different values of a message attribute and perform different actions for each value the ruleset finds.

A ruleset is a text file located in the ~/rules directory that contains one or more rules. In general, each ruleset can be associated with one or more Scalix routes. However, the virus scanning ruleset applies to all routes. See "Creating a Virus Scanning Ruleset (ALL-ROUTES.VIR)" on page 208 to create a virus scanning ruleset.

See "Message Delivery Rulesets" on page 80 for more information about rules and rulesets.

Method of Operation

All incoming messages to Scalix users route through the Service Router. If you configure the Service Router to perform virus-scanning tasks, the Service Router instructs the mapper script (omvscan.map) to either "SCAN" or "SCAN_AND_CLEAN" message attachments.

The mapper script contains information that invokes the McAfee, Trend Micro, or ClamAV command-line interface (uvscan for McAfee VirusScan, iscan for Trend Micro InterScan VirusWall, and clamdscan for ClamAV). Depending on the virus scanning software you configure in the omvscan.cfg file, the relevant command is executed to scan the attachment and the result of the scan is returned to the Service Router.

If the mapper script determines that a virus exists in the attachment, the Service Router refers to the message delivery rulesets defined in the ALL-ROUTES.VIR configuration file. ALL-ROUTES.VIR determines whether the message attachment must be repaired and forwarded, or discarded. ALL-ROUTES.VIR also determine whether notifications must be sent.

Configuring Scalix Virus Protection

The following information lists the files you need to modify to enable virus protection on Scalix Server.

Creating a Virus Scanning Ruleset (ALL-ROUTES.VIR)

You must create the ALL-ROUTES.VIR file in the ~/rules directory in order to control virus protection on the Scalix Server. This file is a Message Delivery Ruleset file which applies to all routes.

There are two message attributes that you can use in virus scanning rules:

- VIRUS-FOUND

  This attribute in a rule causes the Service Router to test each message for the presence of viruses.

- VIRUS-UNCLEANED

  This attribute in a rule causes the Service Router to test each message for the presence of viruses. The Service Router attempts to remove (repair) any infected attachments.

To create the virus scanning ruleset, do the following:

1. Determine whether you want the Service Router to attempt to

   - repair and deliver infected attachments
   - prevent the delivery of infected messages.

   Selecting one of the above actions determines which of the two virus scanning message attributes you use in the virus scanning ruleset.

2. Create a text file containing the virus scanning rules you want to use. Each rule is a single line of text as shown below:

   message-attribute=mvalue action-attribute=avalue action-attribute=avalue ...

   `message-attribute` is either `VIRUS-FOUND` or `VIRUS-UNCLEANED`. `mvalue` is a numerical value specifying the number of viruses detected/uncleaned. Enter `0` to indicate none, or enter `1` to indicate one or more.

   `action-attribute` and `avalue` can be one of the following:

   ACTION=ALLOW

   ACTION=DISCARD

   ACTION=REJECT

   ACTION=DEFER

   ACTION=RETURN

3. Save the text file as **ALL-ROUTES.VIR** in the `~/rules/` directory.

4. Restart the Service Router:

   ```
   omoff -s sr
   omon -s sr
   ```

   Completing this procedure causes the Service Router to test and process all messages on all routes using the criteria you specify in the **ALL-ROUTES.VIR** file.

Examples

1 In the example below, an infected message is discarded and a notification message containing the "NOTIFY" text is sent to the sender of the infected message.

```
VIRUS-FOUND=1 ACTION=DISCARD NOTIFY="<rejection message>"
VIRUS-FOUND=0 ACTION=ALLOW
```

2 The example below shows how to configure the **ALL-ROUTES.VIR** file to repair the message and notify the sender of an infected attachment:

```
VIRUS-UNCLEANED=1 ACTION=REJECT NDN-INFO=!ndninfo.txt
```

If the virus software cannot repair the attachment, Scalix discards the message and a non-delivery notification containing the text in the `~/rules/ndninfo.txt` file is sent to the sender.

3 If the virus software repairs the attachment, Scalix delivers the message (and attachment) to the recipient, but a notification message is sent to the originator warning that a virus was found in the attachment.

```
VIRUS-UNCLEANED=0 VIRUS-FOUND=1 ACTION=ALLOW NOTIFY=
"<found/cleaned msg>".
```

Note Each rule in the ALL-ROUTES.VIR file must be on a single line, nor can the file have any blank lines.

omvscan.map

omvscan.map is the virus scanning mapper script that links Scalix and the following third-party virus scanning applications:

- McAfee VirusScan for Linux
- Trend Micro InterScan VirusWall
- Clam Anti-Virus

The Scalix installation Wizard installs omvscan.map in the /opt/scalix/examples/general directory, However, you must copy omvscan.map into the ~/rules directory to enable the script and virus protection on Scalix.

Note Make sure that the file is owned by the "root" user, the "scalix" group, and has "555" permissions.

The omvscan.map is enabled when the Service Router process begins (startup). The script remains active (enabled) until the Service Router is shutdown. If you configure auxiliary Service Router processes, each Service Router process starts its own instance of omvscan.map.

omvscan.cfg

The **omvscan.cfg** configuration file defines the anti-virus application to use to scan messages and defines the various options to be used by the anti-virus application. omvscan.cfg is in the ~/sys directory.

The following information displays the default omvscan.cfg file:

```
[GENERAL]
ANTI_VIRUS_ENGINE="ClamAV"
OMAV_LOGFILE=~/logs/omvscan.log
# 0 is off, 1 is ERRORS, 2 is ERRORS & WARNINGS, 3 is same as 2 + DEBUG
OMAV_LOGLEVEL=0

[Trend Micro InterScan VirusWall]
TREND_ENGINE=/opt/trend/ISBASE/IScan.BASE/vscan
TREND_SCAN_OPTIONS='-p/etc/iscan -v0 -za'
TREND_CLEAN_OPTIONS='-p/etc/iscan -v0 -za -c'

TREND_LOGPFX=$(omrealpath '~/tmp/trendvs.log')
TREND_USE_LOCKING=no
TREND_LOCK_FILE=trendvs.lock

[McAfee Virus Scan]
MCAFEE_ENGINE=/usr/local/bin/uvscan
MCAFEE_SCAN_OPTIONS='--secure --noboot --mime'

MCAFEE_CLEAN_OPTIONS='--secure --noboot --mime --norename -c'
MCAFEE_LOGPFX=$(omrealpath '~/tmp/mcafee.log')
MCAFEE_USE_LOCKING=no
MCAFEE_LOCK_FILE=mcafee.lock

CLAMAV_ENGINE=/usr/bin/clamdscan
CLAMAV_SCAN_OPTIONS='--stdout'
CLAMAV_CLEAN_OPTIONS='--stdout'

CLAMAV_LOGPGX=$(omrealpath '~/tmp/clamav.log')
CLAMAV_USE_LOCKING=no
CLAMAV_LOCK_FILE=clamav.lock
```

The configuration file includes a "[GENERAL]" section followed by application-specific sections. The options you configure in the "[GENERAL]" section apply only to Scalix and not to any virus-scanning application. The following table lists the parameters in the omvscan.cfg file.

Parameter	Option
ANTI_VIRUS_ENGINE	Specifies the virus scanning engine to use, and must correspond to one of the section names, as follows: [Trend Micro InterScan VirusWall] [McAfee Virus Scan] [ClamAV]
OMAV_LOGFILE	Specifies the file to which the omvscan.map script outputs logging information. NOTE: The Service Router also logs information to various other Scalix log files. You can set Service Router logging using the `omconflvl` command.
OMAV_LOGLEVEL	Specifies the amount of logging information that the mapper script (omvscan.map) outputs: 0: No logging (default) 1: Error conditions only 2: Warnings and errors 3: Debug information (such as responses from the scanning application), and warnings and errors. This is the maximum logging level.

Parameter	Option
{*product*}_ENGINE	Specifies the location of the command-line virus scanning application.
{*product*}_SCAN_OPTIONS	Specifies the scanning options sent to the virus scanning application.
{*product*}_CLEAN_OPTIONS	Specifies the cleaning (repair) options sent to the virus scanning application.
{*product*}_LOGPFX	Specifies the name of the log file to which the temporary output from a virus-scanning application is stored (when the application is invoked to scan or clean a file). This log file is parsed by omvscan.map to get information (such as the name of the virus) if an attachment is infected.
{*product*}_USE_LOCKING	Specifies whether to force omvscan.map to pause while another instance of omvscan.map is already using the virus scanning application to perform a request. All three third-party anti-virus applications can have multiple instances operating at the same time, respectively. For increased Scalix Server performance, the default for this parameter is set to "NO".
{*product*}_LOCK_FILE	If you enable virus-scanning application locking, this parameter specifies the name of the lock file used by the one or more instances of omvscan.map (which are sharing access to the virus-scanning application).

If you install virus scanning software to its default location, you will likely only have to modify "ANTI_VIRUS_ENGINE" parameter.

General.cfg options

The following modification to the `general.cfg` file is useful when setting up virus protection:

```
SR_VS_IGNORE_ITEM_TYPES
```

You can create a colon-separated list of filecodes to exclude from virus scanning. For example, setting this parameter to `1166:1167` excludes Scalix Distribution lists and Text file (**.txt**) attachments from scanning. This can increase Service Router performance.

SMTP Authentication and Anti-spam Protection

Scalix supports SMTP authentication to allow accurate identification of the users of the SMTP service. In addition, Scalix allows you to configure anti-spamming measures to prevent abuse of the Scalix system.

Both of these security measures are implemented as part of the SMTP Relay (omsmtpd), and are configured by adding entries to the SMTP Relay configuration file:

~scalix/sys/smtpd.cfg.

To configure how the SMTP Relay manages incoming connections, you must specify an action that the SMTP Relay performs in response to an event for each address or address pattern.

When an event occurs, the SMTP Relay checks the relevant entries in the configuration file for matching event/pattern entries. The check is done sequentially, from top to bottom.

When it finds the first match, the SMTP Relay takes the action specified. If the SMTP Relay does not find a match, it processes the message normally.

The default configuration file included with Scalix causes the SMTP Relay to accept all relay attempts from hosts in the local domain, and reject all unauthenticated relay attempts from outside the local domain.

The following table lists possible values for the options in the SMTP Relay configuration file.

Event	Description
SUBMIT	An attempt is made to submit a message from the host specified in *pattern*. If the *action* specified is `Header`, the wording of the header inserted is: X-Scalix-Suspicious-Host: *hostname-or-IP-address*
ANONYMOUS	An attempt is made to submit a message sent without authentication or after a failed authentication. If the *action* specified is `Header`, the wording of the header inserted is: X-Scalix-Anonymous-Message: from *hostname at date*
AUTH_SUCCESS	An attempt is made to submit a successfully authenticated message. Normally only used with the `Accept` and `Header` actions. If the *action* specified is `Header`, the wording of the header inserted is: X-Scalix-Authenticated-Sender: *email-address; authenticated by hostname at date*
AUTH_MISMATCH	An attempt is made to submit a message which was successfully authenticated, but the originator name does not match the authenticated user name. If the *action* specified is `Header`, the wording of the header inserted is: X-Scalix-Authentication-Mismatch: Message originated from *email-address*, but authenticated identity is *email-address*
RELAY	An attempt is made to relay a message through the SMTP Relay. Normally, you would specify all local hosts in the associated *pattern*, so that they can all send messages to any external host, and any external host can send messages to the local hosts. A relay attempt from the host on which the SMTP Gateway is running is always accepted. The SMTP Relay always inserts a standard `Received:` header in the message, so a `Header` *action* is not required.
ORIGINATOR	An attempt is made to submit a message from a user whose e-mail address matches the *pattern* specified. Use this event to block mail from known sources of spam. If the *action* specified is `Header`, the wording of the header inserted is: X-Scalix-Suspicious-Originator: *email-address*
RECIPIENT	An attempt is made to submit a message to a user whose e-mail address matches the *pattern* specified. Use this event to block mail to nonexistent addresses. If the *action* specified is `Header`, it will be ignored.
SMTPFILTER	SMTPFILTER has only one state and that's TRUE. So, if you add the following line: SMTPFILTER=TRUE to the ~/sys/smtpd.cfg file, it will cause the SMTP Relay to hand the incoming message off to Sendmail. As a result, inbound messages will be processed through any installed sendmail milters (mail filters) such as Spam Assassin.
Action	**Description**

Event	Description
Accept	The message is unconditionally accepted and processed normally.
Defer	The message is deferred, with a 4XX code. Eventually, the sending host will cease transmitting it, and will reject it.
Discard	The message is accepted but then discarded, with no indication to the sending host that it was not delivered.
Header	The message is accepted, but an extra header is inserted. The wording of the header depends on the *event*.
Reject	The message is rejected, with a 5XX code.
Pattern	Description
Hostname-pattern	The hostname pattern identifies the originating host (or the destination host in the case of the SMTP Relay event). It is used for all events except `ORIGINATOR` and `RECIPIENT`. Possible values are described in the `Hostname Pattern` section.
Email-address-pattern	The e-mail address pattern identifies the source or destination e-mail address, used only for the `ORIGINATOR` and `RECIPIENT` events. For example: `*@*.spam.net` matches `iama.spammer@lotsof.spam.net.` * matches all e-mail addresses.

Note that all actions can be prefixed by the string Log_ to cause the action to be recorded in the Scalix log file. In addition, SMTP Relay information is logged in the Audit Log. See the Audit Log configuration file for more information.

The following information shows the ~scalix/sys/smtpd.cfg configuration file. You configure SMTP Authentication and Anti-spam protection in the second part of the smtpd.cfg file (format: event action pattern pattern ...).

```
##############################################################################
# SMTP Relay Configuration
# ##########################################################################
#
# For details please see Scalix Administration Guide
#
##############################################################################

##############################################################################
# Relay Configuration
# ###################
#
...

##############################################################################
# Authentication and Anti-SPAMming Measures
# ##########################################################################
#
# Each line is of the form:
# EVENT ACTION PATTERN PATTERN...
# When an event happens the SMTP Relay checks for a matching event/pattern
# sequentially in this file. When it finds the first match, it takes the
```

```
        # action specified.
        #
        # ######
        # EVENTS
        # ######
        #

# AUTH_SUCCESS      An attempt is made to submit a successfully
#                   authenticated message.
# AUTH_MISMATCH     An attempt is made to submit a successfully
#                   authenticated message but the originator name does not
#                   match the authenticated name.
# ANONYMOUS         An attempt is made to submit a message sent without
#                   authentication or after failed authentication.
# SUBMIT#           An attempt is made to submit a message from the host
#                   specified in pattern
# RELAY             An attempt is made to relay a msg through the SMTP
#                   Relay
# ORIGINATOR        An attempt is made to submit a message from a user
#                   whose email address matches pattern
# RECIPIENT         An attempt is made to submit a message to a user whose
#                   email address matches pattern



        #
        # #######
        # ACTIONS
        # #######
        #

# Accept          The message is unconditionally accepted and processed normally.
#
# Defer           The message is deferred with a 400 code
# Discard         The message is accepted but then discarded
# Header          The message is accepted, but an extra header is inserted.
# Reject          The message is rejected with a 500 code


        # If Log_ added to the start of an action, then the action is also
        recorded
        # in the SMTP Relay log file.
        #
        # ########
        # PATTERNS
        # ########
        #
        # Hostname Patterns
        # - an IP address, eg 123.234.132.231
        # - an IP subnet and mask, eg 123.234.200.0/255.255.240.0
        # - a hostname, eg bert.loc.co.uk
```

```
# - the end of a domain, eg .spammer.net
# - the start of a domain, 123.234.
# - the keyword ALL matches all hosts
# - the keyword LOCAL matches all hosts that do not contain a .
#
# Email Patterns - used by ORIGINATOR and RECIPIENT
# - *@*.spam.net
#
##############################################################################
RELAY accept domain
RELAY reject ALL
#
# extra rules to prevent open relay usage
RECIPIENT Reject *@*@*
RECIPIENT Reject *%*
RECIPIENT Reject *!*
```

There can be several configuration entries for the same event, but only one of them applies to any particular message. For any event, the SMTP Relay scans all configuration entries (from top to bottom) and looks for the first match. Any other configuration entries for this event are ignored.

Note the following:

- AUTH_MISMATCH: this event is an attempt to submit a message which was successfully authenticated but the originator name (FROM: in the RFC 822 header) did not match the authenticated user name.
- Header: In this action, an extra header is inserted into the message. The header name and the value are fixed and depend on the event type.
- Defer: In this action, an SMTP 400 code is returned to the sending server. This means that the message remains on the sending server and is repeatedly retried until it times out and is rejected by the submitting server. This means that the SPAM message occupies disk space on the sending host, but might cause problems for uninvolved third parties.
- Reject: In this action, the message is rejected and an SMTP 500 code is returned to the sending server. For most positively identified SPAM originators and recipients, this is the preferred action since it requires little processing power.
- If debug logging is enabled and any of the action keywords is prefixed with Log_, this action will also be recorded in the SMTP Relay log file, ~scalix/tmp/smtpd.log. You enable debug logging by adding the line debug_log=true to smtpd.cfg.
- Hostname patterns should be used for the ANONYMOUS, AUTH_SUCCESS, AUTH_MISMATCH, RELAY and SUBMIT events.
- If a hostname cannot be looked up in the DNS, it will not match a domain name pattern or an explicit hostname.
- A subnet and mask separated by a / (for example, 15.145.200.0/255.255.240.00) will match all IP addresses in the 15.145.200.0 to 15.145.207.255 range. Note that the mask need not correspond to a "real" subnet.
- A string that begins with an @ character is treated as an NIS (YP) netgroup name. A host-name is matched if it is a host member of the specified netgroup.
- A string that begins with a / character is treated as a file name. A hostname or address is matched if it matches any hostname listed in the named file. The file format is zero or more lines with zero or more hostname patterns separated by white space.

- E-mail address patterns should be used for ORIGINATOR and RECIPIENT events.

How To Prevent Message Spoofing From Internal Hosts

The following example shows an example of a person in the intranet using SMTP commands to send a message to the server and appear to be a user they are not. For instance, you can telnet to a Scalix Server named scalix1, simulate being the user bob, and send a message to tom:

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
ehlo scalix1
250-scalix1.pwd.scalix.com Hello scalix1.pwd.scalix.com [15.145.204.43],
pleased to meet
you
250-AUTH LOGIN PLAIN DIGEST-MD5 ANONYMOUS CRAM-MD5
250-AUTH=LOGIN
250-DSN
250 8BITMIME
mail from: bob@scalix.com
250 bob@scalix.com... Sender ok
rcpt to: tom@scalix1.pwd.scalix.com
250 Ok
data

354 Enter mail, end with "." on a line by itself (relay)
subject: an important meeting
Please attend the meeting taking place in the boardroom at 9am tomorrow
.
250 Ok
```

The message received by tom appears to be legitimate:

```
Return-Path: <bob@scalix.com>
Received: from scalix1 (scalix1.pwd.scalix.com 15.145.204.43)
by scalix1.pwd.scalix.com via ESMTP; Tue, 17 Apr 2001 14:21:37 +0100 (BST)
Date: Tue, 17 Apr 2001 14:21:40 +0100
From: bob@scalix.com
Sender: bob@scalix.com
Message-ID: <1642.987513700.scalix1.pwd.scalix.com@MHS>
Subject: an important meeting
MIME-Version: 1.0
Content-Type: text/plain;charset="US-ASCII"
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline
Please attend the meeting taking place in the boardroom at 9am tomorrow
```

To prevent such messages from being accepted by the SMTP Relay, you can configure an ANONYMOUS event which instructs the SMTP Relay what to do when an attempt is made to submit a message without authentication (or which failed authentication).

In the current example, you want to reject all anonymous connections from clients in your domain. However, we have a server set up to relay messages, scalixopp.pwd.scalix.com, and it will try to connect anonymously, so you need to allow for this in your configuration:

```
ANONYMOUS Header scalixopp.pwd.scalix.com
ANONYMOUS Reject .pwd.scalix.com
ANONYMOUS Accept ALL
```

Note that the example asks the SMTP Relay to insert a header in anonymous messages which it relays to users from scalixopp. This header takes the following form (where sender is the message sender and addr is the IP address of the sending host):

```
X-Scalix-Anonymous-Message: from sender at addr
```

(In the current example, the addr will be 15.145.205.23 which is the IP address of scalixopp.pwd.scalix.com.)

Alternatively, you can set up your relaying servers to be within a certain IP range and specify the range using the IP subnet and mask.

When you try using a telnet session to make the message appear to be from "bob@scalix.com", the second configuration line is executed and you are asked to authenticate.

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.

Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
ehlo scalix1
250-scalix1.pwd.scalix.com Hello scalix1.pwd.scalix.com [15.145.204.43],
pleased to meet you
250-AUTH LOGIN PLAIN DIGEST-MD5 ANONYMOUS CRAM-MD5
250-AUTH=LOGIN
250-DSN
250 8BITMIME
mail from: bob@scalix.com
530 Authentication required
...
```

When you send a message tom on scalix2.pwd.scalix.com. It gets relayed via scalixopp, so the SMTP Relay executes the first configuration line and you receive the message with an extra header, showing that this message came from scalixopp (15.145.205.23) and is unauthenticated:

```
Return-Path: <tom@scalix2.pwd.scalix.com>
Received: from scalixopp.pwd.scalix.com (scalixopp.pwd.scalix.com
15.145.205.23)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Wed, 18 Apr 2001 16:31:04 +0100 (BST)
Received: from scalixopp.pwd.scalix.com (root@localhost)
by scalixopp.pwd.scalix.com (8.9.3 (PHNE_18546)/8.9.3 SMKit7.01 Scalix)
with
ESMTP id QAA19330
for <tom@scalix1.pwd.scalix.com>; Wed, 18 Apr 2001 16:31:03 +0100 (BST)
Received: from scalix2.pwd.scalix.com (scalix2.pwd.scalix.com
```

```
15.145.204.249)
by scalixopp.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Wed, 18 Apr 2001 16:31:04 +0100 (BST)
Received: from scalix2.pwd.scalix.com (root@localhost)
by scalix2.pwd.scalix.com (8.8.6 (PHNE_17190)/8.8.6) with ESMTP id
QAA03025
for <tom@scalix1>; Wed, 18 Apr 2001 16:31:03 +0100 (BST)
Received: from joyford3 (scalixpwdl86.pwd.scalix.com 15.145.205.60)
by scalix2.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Wed, 18 Apr 2001 16:31:03 +0100 (BST)
Date: Wed, 18 Apr 2001 16:31:02 +0100
From: "tom" <tom@scalix2.pwd.scalix.com>
Sender: "tom" <tom@scalix2.pwd.scalix.com>
To: tom@scalix1.pwd.scalix.com
Message-ID: <001401c0c81c$938a7ca0$62cc910f@pwd.scalix.com>
Subject: some headers
X-MSMail-Priority: Normal
X-Priority: 3
X-Scalix-Anonymous-Message: from <tom@scalix2.pwd.scalix.com> at
15.145.205.23
X-Mailer: Microsoft Outlook Express 5.50.4133.2400
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4133.2400
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="----=_NextPart_000_0011_01C0C824.F520F6D0"
...
```

If you have several hosts set up to relay messages around your intranet, you will need to include them in the ANONYMOUS line to allow them to connect without authenticating.

Remember that for each connection, the SMTP Relay reads down through the configuration file from the top and execute the first line that matches. So the selective line:

```
ANONYMOUS Header scalixopp.pwd.scalix.com
```

must come before the more global line:

```
ANONYMOUS Reject .pwd.scalix.com
```

Verifying the Identity of a Sender

You might want the SMTP Relay to accept all successfully authenticated messages, but for tracking purposes, you want a header added to messages from your domain (pwd.scalix.com), to confirm the identity of the authenticated sender and the address of the sending client/host.

The event, in this case, is AUTH_SUCCESS, the action is Header and the pattern is .pwd.scalix.com. Therefore, you can add following lines to the configuration file:

```
AUTH_SUCCESS Header .pwd.scalix.com
AUTH_SUCCESS Accept ALL
```

This instructs the SMTP Relay to add an extra header (X-Scalix-Authenticated-Sender:) to authenticated messages from any host ending in .pwd.scalix.com and to accept authenticated messages from any other hosts (LOCAL or not in the pwd.scalix.com domain).

For example, you can add the above code to smtpd.cfg on the server, scalix1, and use Outlook on a separate PC (IP Address = 15.145.205.60) to send a message from kelly on scalix1 to Fred on scalix1. Below is the message received by Fred, showing the added header with kelly's address and the IP address of the connecting machine:

```
Return-Path: <kelly@scalix1.pwd.scalix.com>
Received: from scalix1.pwd.scalix.com (root@localhost)
by scalix1.pwd.scalix.com (8.8.6 (PHNE_17190)/8.8.6) with ESMTP id
LAA03978
for <fred@scalix1>; Wed, 18 Apr 2001 11:21:43 +0100 (BST)
Received: from joyford3 (scalixpwdl86.pwd.scalix.com 15.145.205.60)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Wed, 18 Apr 2001 11:21:43 +0100 (BST)
Date: Wed, 18 Apr 2001 11:21:41 +0100
From: kelly@scalix1.pwd.scalix.com
Sender: kelly@scalix1.pwd.scalix.com
To: fred@scalix1.pwd.scalix.com
Message-ID: <002f01c0c7f1$5cb78270$62cc910f@pwd.scalix.com>
Subject: hi
X-MSMail-Priority: Normal
X-Priority: 3
X-Scalix-Authenticated-Sender: <kelly@scalix1.pwd.scalix.com> at
15.145.205.60
X-Mailer: Microsoft Outlook Express 5.50.4133.2400
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4133.2400
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="----=_NextPart_000_002C_01C0C7F9.BE468290"
...
```

AUTH_SUCCESS should only be used with the Header action to insert the additional header verifying the user's identity.

Handling Messages With Sender/Authenticated User Mismatch

We looked at how a "spoofer" might try to send an unauthenticated message. The following information describes how to prevent the "spoofer" from connecting to `scalix1`, authenticating successfully, and sending a fake message to kelly.

To prevent this, use the AUTH_MISMATCH event. AUTH_MISMATCH describes an attempt to submit a message which was successfully authenticated, but the originator name (FROM:) did not match the authenticated user name. Such messages should be rejected.

For example, you can add the following lines to **smtpd.cfg**:

```
AUTH_MISMATCH Reject LOCAL
AUTH_MISMATCH Header scalix2.pwd.scalix.com
AUTH_MISMATCH Reject ALL
```

The first line causes all authenticated messages from the local host (no . in the address) to be rejected (with a SMTP 500 response), if the sender does not match the authenticated user.

The second line causes a header to be added to any similarly deficient message from the host, scalix2.pwd.scalix.com. The message is not rejected.

The third line causes any other message with this defect to be rejected.

When you telnet to scalix1, authenticate as tom, and try to send a message as bob:

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
Connection closed by foreign host.
root@scalix1[] telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
ehlo scalix1
250-scalix1.pwd.scalix.com Hello scalix1.pwd.scalix.com [15.145.204.43],
pleased to meet you
250-AUTH LOGIN PLAIN DIGEST-MD5 ANONYMOUS CRAM-MD5
250-AUTH=LOGIN
250-DSN
250 8BITMIME
auth login
334 VXNlcm5hbWU6
bXIgYWRtaW4=
334 UGFzc3dvcmQ6
YWRtaW4=
235 Authentication successful
mail from: bob@scalix.com
530 Authentication mismatch
```

Scalix found the mismatch and the connection rejected with a SMTP 530 error code.

When you try sending the message by performing a telnet from scalix2 to scalix1, the following occurs:

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
ehlo scalix2
250-scalix1.pwd.scalix.com Hello scalix2.pwd.scalix.com [15.145.204.249],
pleased to meet you
250-AUTH LOGIN PLAIN DIGEST-MD5 ANONYMOUS CRAM-MD5
250-AUTH=LOGIN
250-DSN
250 8BITMIME
auth login
334 VXNlcm5hbWU6
bXIgYWRtaW4=
334 UGFzc3dvcmQ6
YWRtaW4=
235 Authentication successful
mail from: bob@scalix.com
250 bob@scalix.com... Sender ok
```

```
rcpt to: kelly@scalix1
250 Ok
data
354 Enter mail, end with "." on a line by itself (relay)
subject: an important meeting
Please attend the boardroom meeting tomorrow, 9am
bob
.
250 Ok
```

The message is accepted, but there is a header highlighting the discrepancy:

```
Return-Path: <bob@scalix.com>
Received: from scalix2 (scalix2.pwd.scalix.com 15.145.204.249)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Wed, 18 Apr 2001 17:51:59 +0100 (BST)
Date: Wed, 18 Apr 2001 17:51:59 +0100
From: bob@scalix.com
Sender: bob@scalix.com
Message-ID: <4752.987612719.scalix1.pwd.scalix.com@MHS>
Subject: an important meeting
X-Scalix-Authentication-Mismatch: originator bob@scalix.com
authenticated as tom at 15.145.204.249
MIME-Version: 1.0
Content-Type: text/plain;
charset="US-ASCII"
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline
Please attend the boardroom meeting tomorrow, 9am
bob
```

The IP address is that of scalix2.

Restricting Who Can Use This Server To Relay Mail

It is very important to prevent outside hosts using your server to relay large quantities of unwanted mail to other internal and external hosts. You can use configuration entries for RELAY events to tell the SMTP Relay what to do when a host, which matches the pattern, attempts to send a message through the SMTP Relay to a Sendmail recipient on another server.

Normally, all local hosts should be included so that they will be allowed to send messages to any external host. You can also list here any external hosts which are allowed to use this server to relay.

For example, note the two entries in the default file. These are automatically inserted by Scalix, including the domain of this host:

```
RELAY Accept domain

RELAY Reject ALL
```

This default setup means that any attempt to use this server to relay a message will only be successful if the sending server is in the same domain as this server. As the SMTP Relay always inserts a standard Received: header into the message, the Header action does not make sense for a RELAY event and will be ignored.

Note that SMTP Relay cannot block relay attempts through the Internet Gateway (the message goes into Scalix and is relayed using Scalix routes). See “Internet Mail Gateway” on page 29 for more information.

Blocking Mail From Certain Hosts

Configuration entries for SUBMIT events describe what the SMTP Relay does when a host matching one of the given patterns attempts to submit a message. You can use lines like the examples below to block/log message submission from hosts which are known or suspected of sending large amounts of unwanted mail (spam):

```
SUBMIT Reject known.spammer.net

SUBMIT Log_Reject another.spammer.net

SUBMIT Header possible.spammer.net

SUBMIT Accept ALL
```

the SMTP Relay looks at the address in the MAIL FROM: line of the message and looks through the SUBMIT lines in the configuration file to see if the address matches any of those specified. If the address is known.spammer.net, an SMTP 500 code is returned and the message is not accepted.

If the address is another.spammer.net, an SMTP 500 code is returned, the message is not accepted and, if debug logging is enabled, this action is logged in the log ~scalix/tmp/smtpd.log file.

If the address is possible.spammer.net, the message will be accepted but the following header will be added to the RFC 822 message header to indicate that the message was from a suspect host:

```
X-Scalix-Suspicious-Host: IP address
```

If the message is submitted by any host other than those identified in the lines above, it is accepted.

For example, you determine that a host (scalixopp) might be sending unwanted mail, and you want the SMTP Relay to add a header to any message from that host instructing the recipient that the sender is not trusted. Also, you are certain that scalix2 is sending SPAM and you want to reject connections from scalix2 and have the rejection logged in the SMTP Relay’s log file.

Add the following lines to scalix1’s Relay configuration file:

```
SUBMIT Header scalixopp.pwd.scalix.com

SUBMIT Log_Reject scalix2.pwd.scalix.com
```

However, you know that messages from scalix2 will be relayed to scalix1 by scalixopp. The following shows what happens when you send a message from tom on scalix2 to Fred on scalix1:

```
Return-Path: <tom@scalix2.pwd.scalix.com>
Received: from scalixopp.pwd.scalix.com (scalixopp.pwd.scalix.com
15.145.205.23)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Thu, 19 Apr 2001 09:27:29 +0100 (BST)
Received: from scalixopp.pwd.scalix.com (root@localhost)
```

```
by scalixopp.pwd.scalix.com (8.9.3 (PHNE_18546)/8.9.3 SMKit7.01 Scalix)
with
ESMTP id JAA20532
for <fred@scalix1.pwd.scalix.com>; Thu, 19 Apr 2001 09:27:28 +0100 (BST)
Received: from scalix2.pwd.scalix.com (scalix2.pwd.scalix.com
15.145.204.249)
by scalixopp.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Thu, 19 Apr 2001 09:27:28 +0100 (BST)
Received: from scalix2.pwd.scalix.com (root@localhost)
by scalix2.pwd.scalix.com (8.8.6 (PHNE_17190)/8.8.6) with ESMTP id
JAA03131
for <fred@scalix1>; Thu, 19 Apr 2001 09:27:26 +0100 (BST)
Received: from joyford3 (scalixpwdl86.pwd.scalix.com 15.145.205.60)
by scalix2.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Thu, 19 Apr 2001 09:27:27 +0100 (BST)
Date: Thu, 19 Apr 2001 09:27:26 +0100
From: "tom" <tom@scalix2.pwd.scalix.com>
Sender: "tom" <tom@scalix2.pwd.scalix.com>
To: fred@scalix1.pwd.scalix.com
Message-ID: <001401c0c8aa$90ca33a0$62cc910f@pwd.scalix.com>
Subject: a message
X-MSMail-Priority: Normal

X-Priority: 3
X-Scalix-Suspicious-Host: 15.145.205.23
X-Mailer: Microsoft Outlook Express 5.50.4133.2400
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4133.2400
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="----=_NextPart_000_0011_01C0C8B2.F2455DA0"
...
```

A header is inserted to show that this message was sent (or relayed) to Fred by scalixopp (15.145.205.23), one of the suspect hosts.

If you try sending a message directly from scalix2 to scalix1:

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
550 Denied
Connection closed by foreign host.
```

The connection is refused immediately. Note the rejection logged in the smtpd.log file on scalix1.

```
Rejected connection from 15.145.204.249
```

where 15.145.204.249 is the IP address of scalix2.

Blocking Mail From Specific Senders

The ORIGINATOR event describes an attempt to send a message from a user whose e-mail address matches a pattern. We can use this event to block mail coming from known spammers.

Here are some example lines:

```
ORIGINATOR Log_Reject spam@advert.com

ORIGINATOR Discard spam@blast.net

ORIGINATOR Defer spam*@*

ORIGINATOR Accept ALL
```

The first line rejects any message from spam@advert.com and logs the rejection in smtpd.log. The sending host will get a SMTP 500 response. The second line accepts messages form spam@blast.net but discards immediately discards them.

The third line defers the delivery of any messages from addresses matching spam*@*. The sending hosts receives an SMTP 400 response and the messages is stored on the sending host. The submission is attempted until the sending host rejects the messages.

If you set the header action, the inserted header takes the following form:

```
X-Scalix-Suspicious-Originator: email address
```

To flag any messages as suspicious from users with the scalix2.pwd.scalix.com domain in their address, add the following line to smtpd.cfg on scalix1:

```
ORIGINATOR Header *@scalix2.pwd.scalix.com
```

When you use an Internet client to send a message from tom on scalix2 to Fred on scalix1, the message has the extra header underlined below:

```
Return-Path: <tom@scalix2.pwd.scalix.com>
Received: from scalixopp.pwd.scalix.com (scalixopp.pwd.scalix.com
15.145.205.23)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Thu, 19 Apr 2001 14:30:16 +0100 (BST)
Received: from scalixopp.pwd.scalix.com (root@localhost)
by scalixopp.pwd.scalix.com (8.9.3 (PHNE_18546)/8.9.3 SMKit7.01 Scalix)
with
ESMTP id OAA18566
for <tom@scalix1.pwd.scalix.com>; Thu, 19 Apr 2001 14:30:15 +0100 (BST)
Received: from scalix2.pwd.scalix.com (scalix2.pwd.scalix.com
15.145.204.249)
by scalixopp.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Thu, 19 Apr 2001 14:30:15 +0100 (BST)
Received: from scalix2.pwd.scalix.com (root@localhost)
by scalix2.pwd.scalix.com (8.8.6 (PHNE_17190)/8.8.6) with ESMTP id
OAA03265
for <tom@scalix1>; Thu, 19 Apr 2001 14:30:14 +0100 (BST)
Received: from joyford3 (scalixpwdl86.pwd.scalix.com 15.145.205.60)
by scalix2.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTP; Thu, 19 Apr 2001 14:30:15 +0100 (BST)
Date: Thu, 19 Apr 2001 14:30:14 +0100
From: "tom" <tom@scalix2.pwd.scalix.com>
Sender: "tom" <tom@scalix2.pwd.scalix.com>
To: tom@scalix1.pwd.scalix.com
Message-ID: <001401c0c8d4$ddc0b5b0$62cc910f@pwd.scalix.com>
Subject: see the headers
X-MSMail-Priority: Normal
X-Priority: 3
```

```
X-Scalix-Suspicious-Originator: <tom@scalix2.pwd.scalix.com> at
X-Mailer: Microsoft Outlook Express 5.50.4133.2400
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4133.2400
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="----=_NextPart_000_0011_01C0C8DD.3F55A940"
...
```

Blocking Mail For Specific Recipients

Use the RECIPIENT event to manage messages sent to addresses that do not exist. This option is useful if an individual is no longer part of an organization and you want to stop mail from being delivered to that account without actually removing (deleting) the account.

You can configure RECIPIENT events as follows:

```
RECIPIENT Log_Reject Unknown.User@pwd.scalix.com

RECIPIENT Reject User.HasLeftTheCompany@pwd.scalix.com

RECIPIENT Accept ALL
```

The first line rejects incoming messages for the user Unknown.User@pwd.scalix.com and logs the rejection in smtpd.log.

The second line rejects mail for User.HasLeftTheCompany@pwd.scalix.com.

Microsoft Outlook Security Model

The Outlook E-mail Security Model provides protection against software viruses that users might receive in their Inbox as an attachment file.

To use the security model, create one or more security files on the Scalix Server that contain the configuration parameters described in “Security File Configuration Options” on page 227. You can create one file for all users, or many files for individual users or groups of users.

Note	Scalix Connect for Microsoft Outlook can be used only by Premium users. For more information, see "About Scalix Product Editions".

Using the Outlook Security Model for System-wide Use

The following information describes how to use the Outlook security model for general (system-wide) use:

1. On the Scalix Server, open a new document/file with any Linux text editor application.
2. Enter the security settings described in “Security File Configuration Options” on page 227. To view an example of a security file, see “Security File Example” on page 230.
3. Save the file. You can use any name to identify the security file.
4. Add the following entry in the `~/sys/`**general.cfg** file:

```
UAL_OUTLOOK_SECURITY_FILE=~/sys/security file
```

This setting instructs Outlook to use the security parameters in the file you created in the steps above for all users on the server.

Using the Outlook Security Model for an Individual User

The following information describes how to use the Outlook security model for an individual user:

1. On the Scalix Server, go to the ~/sys/ directory and enter the following command using the common name of the user to which you want to apply security settings:

   ```
   omshowu -n "/CN=common name" -G
   ```

   This command lists (among other items) the Internal User ID of a user.

2. Use the Internal User ID value to identify the configuration file you must edit to apply the security settings. All user configuration files are in the ~/sys/ directory. The file names are derived from the Internal User ID values of the users to which the files respectively belong.

   If the user to which you want to apply the security settings does not have a configuration file:

   - Obtain the Internal User ID of the user by entering the following command:

     ```
     omshowu -n "/CN=common name" -G
     ```

   - Open a new document/file with any Linux text editor application.
   - Save the file to the ~/sys/ directory using the Internal User ID of the user to which you want to apply the security settings.

3. Open a new document/file with any Linux text editor application.
4. Enter the security settings described in "Security File Configuration Options" on page 227. To see an example of a general security file, see "Security File Example" on page 230.
5. Save the file. You can use any name to identify the security file.
6. Add the following entry to the appropriate user configuration file in the ~/sys/ directory.

   ```
   UAL_OUTLOOK_SECURITY_FILE=~/sys/security file
   ```

   This setting instructs Outlook to use the security parameters you created in the steps above for this user only.

   Alternatively, you can enter the following in the user configuration file:

   ```
   UAL_OUTLOOK_SECURITY_FILE=~/sys/outlook.sec
   ```

If you created a system-wide security configuration file (as described in "Using the Outlook Security Model for System-wide Use" on page 225), this setting instructs Outlook to use the security settings in **outlook.sec** for this user, while the security settings for all other users is controlled by the system-wide security configuration file.

If you do not specify a security file in the **general.cfg** file or in the user configuration file(s), Outlook uses the security settings in the *~/sys/***outlook.sec** file. If the **outlook.sec** file does not exist, Outlook uses its default security settings.

Note	You can create groups of users with the same security settings by specifying the same security file in the user configuration files of the users to which you want to apply the security settings.

Security File Configuration Options

The security file you create contains configuration parameters that allow you to determine how the security model operates. The security file can contain the following sections:

- [Level 1 File Extensions]
- [Level 2 File Extensions]
- [Miscellaneous Attachment Settings]
- [Miscellaneous Custom Form Settings]
- [Programmatic Settings]

Make sure the section headings in the security file display as they appear in the list above, including the square brackets ([ ]). You do not have to include a section heading in the security file if you do not need to configure the parameters for that section.

See “Security File Example” on page 230 to view an example of a security file.

Level 1 File Extensions

The following table lists the parameters for the [Level 1 File Extensions] section.

Parameter	Description
Add=	Enables you to add file types to the Level 1 file list. Incoming messages with attachments that match the file type(s) you specify for this parameter are removed from the message. To view a list of file-type extensions that are generally considered “unsafe”, go to: http://support.microsoft.com/support/kb/articles/Q262/6/31.asp
Remove=	Enables you to remove file types from the Level 1 list. Incoming messages with attachments that match the file type(s) you specify for this parameter are retained in the message. When you remove a file type from the Level 1 file list, the file type remains on the Level 2 file list. To remove a file type from both the Level 1 and Level 2 file lists, include Remove=value in both the [Level 1 File Extensions] and [Level 2 File Extensions] sections.

The values you enter for these parameters are file-type extensions without the period. For example, enter Add=bat and not Add=.bat for Batch files. Also if you want to enter multiple file type extensions for a parameter, separate the extensions with a semi-colon. For example Add=bat;doc;rtf;xls.

Level 2 File Extensions

The following table lists the parameters for the [Level 2 File Extensions] section.

Parameter	Description
Add=	Enables you to add file types to the Level 2 file list. Incoming messages with attachments that match the file type(s) you specify for this parameter are removed from the message.
Remove=	Enables you to remove file types from the Level 2 list. Incoming messages with attachments that match the file type(s) you specify for this parameter are retained in the message.

The values you enter for these parameters are file-type extensions without the period. For example, enter Add=bat and not Add=.bat for Batch files. Also if you want to enter multiple file type extensions for a parameter, separate the extensions with a semi-colon. For example Add=bat;doc;rtf;xls.

[Miscellaneous Attachment Settings]

The following table lists the parameters for the [Miscellaneous Attachment Settings] section.

Parameter	Description
ShowLevel1Attachments=	When you set this parameter to TRUE, users can access all attachments in the `[Level 1 File Extensions]` section. When you set this parameter to FALSE, users can only access attachments specified in the Add= parameter of the `[Level 1 File Extensions]` section.
DoNotPromptLevel1Attachments OnSend=	When you set this parameter to TRUE, users do not receive a warning when they send an item containing a Level 1 attachment. After the item is sent, users cannot view or access the attachment. When you set this parameter to FALSE, users receive a warning message.
DoNotPromptLevel1Attachments OnClose=	When you set this parameter to TRUE, users do not receive a warning when they close an item containing a Level 1 attachment. When the item is closed, users cannot view or access the attachment. If you set this parameter to FALSE, users receive a warning message.
AllowActivationOfOLEObjects=	When you set this parameter to TRUE, users can double-click on an embedded attachment, such as Microsoft Excel spreadsheet, and open it in the program. Set this parameter to FALSE to disable this capability. If users are using Microsoft Word as their e-mail editor, users can still open embedded objects even if you set this parameter to FALSE.
ShowOLEPackageObjects=	When you set this parameter to TRUE, users can view all packaged OLE objects (a package is an icon that represents an embedded or linked OLE object). Set this parameter to FALSE to disable this capability. When users open the package icon, the application that created the OLE object executes the OLE object.

Miscellaneous Custom Form Settings

The following table lists the parameters for the [Miscellaneous Custom Form Settings] section.

Parameter	Description
EnableScriptsInForm=	When you set this parameter to TRUE, users can run scripts using Outlook forms. Outlook allows scripts to run when the script and the layout are contained in the message itself.
ExecutingCustomActionViaOOM=	Specify the action Outlook takes when a program attempts to run a custom action using the Outlook object model. A custom action can be created to reply to a message and circumvent the programmatic send protections. The actions are: Prompt: Users receive a message enabling them to allow or deny the operation. Approve: Users do not receive a prompt and the operation is allowed. Deny: Users do not receive a prompt and the operation is denied.
AccessingItemPropOfaControl OnForm=	Specify the action Outlook takes when a user adds a control to a custom Outlook form and then binds that control directly to any of the Address Information fields. By doing this, code can be used to indirectly retrieve the value of the Address Information field by obtaining the Value property of the control. Prompt: Users receive a message enabling them to allow or deny the operation. Approve: Users do not receive a prompt and the operation is allowed. Deny: Users do not receive a prompt and the operation is denied.

[Programmatic Settings]

The following table lists the parameters for the [Miscellaneous Custom Form Settings] section. The parameters in this section require that you enter one of the following values:

- Prompt: Users receive a message enabling them to allow or deny the operation.
- Approve: Users do not receive a prompt and the operation is allowed.
- Deny: Users do not receive a prompt and the operation is denied.

Parameter	Description
SendingItemsViaOOM=	Specify the action Outlook takes when a program attempts to send mail programmatically using the Outlook object model.
SendingItemsViaCDO=	Specify the action Outlook takes when a program attempts to send mail programmatically using CDO (Collaboration Data Objects).
SendingItemsViaSMAPI=	Specify the action Outlook takes when a program tries to send mail programmatically using Simple MAPI.
AccessingAddrBookViaOOM=	Specify the action Outlook takes when a program attempts to access an address book using the Outlook object model.

Parameter	Description
AccessingAddrBookViaCDO=	Specify the action Outlook takes when a program tries to access the address book using CDO.
ResolvingNamesViaSMAPI=	Specify the action Outlook takes when a program tries to access the address book using Simple MAPI.
AccessingAddrInfoViaOOM=	Specify the action Outlook takes when a program attempts to access address information through Outlook object model (for example, access a recipient field such as “To”).
AccessingAddrInfoViaCDO=	Specify the action Outlook takes when a program attempts to address information through CDO (for example, access a recipient field such as “To”).
OpeningMessagesViaSMAPI=	Specify the action Outlook takes when a program attempts to access a recipient field (such as “To”) using Simple MAPI.
RespondingToMeetingsAndTasks ViaOOM=	Specify the action Outlook takes when a program attempts to send mail programmatically using the Respond method on task requests and meeting requests. This method is similar to the Send method on mail messages.
ExecutingSaveAsViaOOM=	Specify the action Outlook takes when a program attempts to use (programmatically) the Save As command in the File menu to save an item. When an item is saved, a malicious program can search the file for e-mail addresses.
AccessingFormulaPropOfObject InOOM=	Specify the action Outlook takes when users add a Combination or Formula custom field to a custom form, and bind it to an Address Information field. By doing this, code can be used to indirectly retrieve the value of the Address Information field by obtaining the Value property of the field.
AccessingAddrInfoViaUserPropsInOOM=	Specify the action Outlook takes when a program attempts to search mail folders for address information through “UserPropertiesFind” in the Outlook object model.

Security File Example

The following information provides an example of an Outlook security configuration file. The parameters in this example are described in “Security File Configuration Options” on page 227.

```
[Level 1 File Extensions]
Add=bat;vbs
Remove=doc;xls

[Level 2 File Extensions]
Add=zip

[Miscellaneous Attachment Settings]
ShowLevel1Attachments=FALSE
DoNotPromptLevel1AttachmentsOnSend=FALSE
DoNotPromptLevel1AttachmentsOnClose=FALSE
AllowActivationOfOLEObjects=FALSE
ShowOLEPackageObjects=FALSE

[Miscellaneous Custom Form Settings]
EnableScriptsInForm=FALSE
```

```
ExecutingCustomActionViaOOM=PROMPT
AccessingItemPropOfaControlOnForm=PROMPT

[Programmatic Settings]
SendingItemsViaOOM=PROMPT
SendingItemsViaCDO=PROMPT
SendingItemsViaSMAPI=PROMPT
AccessingAddrBookViaOOM=PROMPT
AccessingAddrBookViaCDO=PROMPT
ResolvingNamesViaSMAPI=PROMPT
AccessingAddrInfoViaOOM=PROMPT
AccessingAddrInfoViaCDO=PROMPT
OpeningMessagesViaSMAPI=PROMPT
RespondingToMeetingsAndTasksViaOOM=PROMPT
ExecutingSaveAsViaOOM=PROMPT
AccessingFormulaPropOfObjectInOOM=PROMPT
AccessingAddrInfoViaUserPropsInOOM=PROMPT
```

Chapter 19

Kerberos Authentication

This chapter describes how to use the Kerberos authentication protocol in Scalix environment and includes the following information:

About Kerberos Authentication

Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography. Kerberos uses authentication tickets to authenticate users and/or services.

A Kerberos client can perform secure communications with a Kerberos service if both the client and the service authenticate against a Kerberos Distribution Center (the KDC - the Kerberos Server) and obtain a Ticket Granting Ticket (TGT). The client then requests a service ticket for a specific service.

Therefore, a triangular relationship exists between the client and the KDC, the service and the KDC and between the client and the service. A Kerberos principal is either a client identity or a service identity operating in the Kerberos realm.

In the Scalix environment, you can use secure Kerberos communication with the following Scalix Services:

- Remote Execution Service
- Scalix Administration Console Service
- Scalix UAL Service
- Scalix IMAP Service

You can also configure Single Sign-on authentication with a KDC on the master domain controller that uses Microsoft Active Directory. This allows Scalix users to automatically authentication with the Scalix Server when they login to their Windows domain. See "Single Sign-on Kerberos Authentication" on page 234 for more information.

Configuring Kerberos authentication is optional and not required to use Scalix Services. You do not need to generate a keytab for the Scalix Administration Console Service or the

Remote Execution Service if you plan to use the Scalix Administration Console in Local Mode. See the *Scalix Installation Guide* for more information.

Single Sign-on Kerberos Authentication

Single Sign-on Authentication allows MS Outlook users to access their e-mail using the Kerberos security protocol. This authentication mechanism allows users to login to their local domain in a Microsoft Active Directory® environment and access their e-mail without any further authentication.‘

The Active Directory service is a core component of the Windows operating system. It provides a directory service supporting LDAP, and a Kerberos Key Distribution Center (KDC) to authenticate users. It allows organizations to share and manage information about network resources and users and provides a Single Sign-on environment that integrates with the standard Windows desktop login. In addition, it acts as a single point of management for Windows-based user accounts, clients, servers and applications.

Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography. Kerberos uses authentication tickets to authenticate users and/or services.

A Kerberos client can perform secure communications with a Kerberos service if both the client and the service authenticate against a Kerberos Distribution Center (the KDC - the Kerberos Server) and obtain a Ticket Granting Ticket (TGT). The client then requests a service ticket for a specific service.

Therefore, a triangular relationship exists between the client and the KDC, the service and the KDC and between the client and the service. A Kerberos principal is either a client identity or a service identity operating in the Kerberos realm.

When a user logs into the domain, a request is made for a Ticket, and once authenticated, the user can use the ticket for as long as it remains valid. Server-side Tickets are stored in the **keytab** file. Client-side tickets are stored in a temporary file.

When you launch MS Outlook, the Scalix Connect connector uses the user’s Kerberos credentials to access the Server.

To implement the Single Sign-on environment, you must have:

- Active Directory
- the latest version of Scalix Connect for Microsoft Outlook
- the latest version of Scalix Server
- the ktpass utility from the Microsoft Developer Network support web site

Obtaining ktpass

The ktpass utility creates Kerberos keytab files that are used by Linux Kerberos-based systems to define Key Distribution Center (KDC) hosts and user/service mappings.

ktpass is available from the Windows 2000 resource kit and the Windows Server 2003 installation CD under \Support\Tools\Support.cab.

For more information on this tool, go to the following URL:

http://www.microsoft.com/windows2000/techinfo/planning/security/kerbsteps.asp

You must install the ktpass utility on the Windows domain controller server.

Configuring Single Sign-on

Single Sign-on configuration requires you to make configuration changes to DNS and create an Active Directory user. This user, for Single Sign-on purposes, is actually Scalix Service. You must create an Active Directory "user" for the UAL Scalix Service. When this is complete, you then convert this user/service into a Kerberos Service Principal.

Note	You can also create an Active Directory user for the Scalix IMAP Service if Single Sign-on users in your network use the Evolution e-mail client.

To configure Single Sign-on authentication, do the following steps.

1. On the Domain Controller, go to Start > Programs > Administrative tools > **DNS**.
2. Make sure you have created Forward Lookup Zones for your domains and created Host records for all Scalix Servers in the appropriate Forward Lookup Zone.
3. Under **Forward Lookup Zones**, select a Scalix Server Single Sign-on domain and go to Action > **New Alias**.
4. Enter `scalix-default-mail` in the **Alias name** field, and the fully-qualified name of the Scalix Server with which you are using Single Sign-on in the **Fully qualified name for target host** field (for example, `scalixserver.acme.net`)
5. Click **OK**.
6. Select **Reverse Lookup Zones** and make sure you have created Reverse Lookup Zones for your domain subnets.
7. In subnet in which the Single Sign-on Scalix Server resides, select Action > **New Pointer**.
8. Enter the last two or three digits of the Scalix Server IP address and fully-qualified hostname of the Scalix Server (for example, `scalixserver.acme.net`).
9. Click **OK**.
10. Close the DNS window.
11. Go to Start > Programs > Administrative Tools > **Active Directory Users and Computers**.
12. If it does not already exist, right-click the root domain controller and select New > **Organizational Unit** and name the new unit `Scalix Services`.

    This creates a separate Organizational Unit to contain Scalix Server data.
13. Select the new Scalix Services organizational unit and select Action > New > **User**.
14. Enter `scalix-ual` in the **First Name** field.

    You can also enter the name of the Single Sign-on Scalix Server in the Last Name field. This allows you to identify the keytabs you generate for multiple Scalix Servers.

15 Do not modify the selection (domain) in the pull-down menu.

16 Enter `scalix-ual` in the **User logon name** field.

17 Click **Next**.

The Password window displays.

18 Enter and confirm a password for the user. Make sure that the password you enter is sufficiently complex and that;

- the User must change password at next logon field is not selected
- the User cannot change password field is not selected
- the Password never expires field is selected

19 Click **Next**.

If Microsoft Exchange is installed on the server, the Exchange Mailbox window displays.

20 Clear the **Create an Exchange Mailbox** field.

21 Click **Next**

22 Click **Finish**.

This completes the creation of an Active Directory user that represents the Scalix UAL Service for the Scalix Server.

23 On the domain controller, open a DOS window and change the directory (cd) to the directory that contains ktpass (typically, `c:\Program Files\Support Tools`). See “Obtaining ktpass” on page 234 for more information.

24 To change the Scalix Service account to a Kerberos Service account and generate a keytab, enter:

```
ktpass –princ scalix-ual/scalixservername.domain@REALM –mapuser scalix-
ual -pass password –out path\filename -kvno 3
```

For example:

```
ktpass -princ scalix-ual/scalixserver.acme.net@ACME.NET -mapuser scalix-
ual -pass password -out scalix-ual.keytab -kvno 3
```

Note The -kvno option prevents potential key version mismatches that cause SSO to fail. Setting this value to 3 ensures that keytab version is the same for existing and future users in Active Directory.

25 If you used the Last name field in Step 14 and entered `scalixserver1`, enter:

```
ktpass -princ scalix-ual/scalixserver.acme.net@ACME.NET -mapuser scalix-
ual-scalixserver1 -pass password -out scalix-ual-scalixserver1.keytab
```

You should see the following information indicating that the keytab was successfully created:

```
Successfully mapped scalix-ual/scalixserver.acme.net to scalix-ual.
Key created.
Output keytab to scalix-ual.keytab:
Keytab version: 0x502
```

```
keysize 68 scalix-ual/scalixserver.acme.net@ACME.NET ptype 1
(KRB5_NT_PRINCIPAL)
vno 3 etype 0x1 (DES-CBC-CRC) keylength 8 (0xe6fb762ad01f8a9b)
Account has been set for DES-only encryption.
```

26 Securely copy the keytab to the home directory of the Single Sign-on Scalix Server. You can use a floppy disk or the `scp` command to transfer the keytab.

27 On the Scalix Server, login using your Linux account and then change to `root` user.

28 Merge the keytab you created with the Kerberos system keytab file. Enter:

`ommergekeys /path/filename.keytab`

29 Modify the `/etc/`**krb5.conf** file. Enter:

```
omkrbconf -r REALM -s servername.domain -d domain
```

- `-r` specifies the realm that the Kerberos database controls. For example, if your domain name is acme.com, your realm is ACME or ACME.NET.
- `-s` specifies the fully qualified host name of the Kerberos KDC machine. For Single Sign-on, the KDC is the Domain Controller with Active Directory installed.
- (optional) `-d` specifies the domain name in which the Kerberos Realm operates. If you do not specify a value, the domain name is determined from the current domain.

30 In order for Single Sign-on to operate, the authentication ID for a Scalix Server mailbox must match the domain identity (the ID in Active Directory) for the user. For example, if `jsmith@acme.net` is the User Logon ID for a user in Active Directory, enter the following on the Scalix Server:

```
ommodu –o jsmith -–authid jsmith@ACME.NET
```

The `REALM` information MUST be uppercase.

31 To view the `authid` value (`-o`) for a user, enter:

```
omshowu "Joe Smith/mailnode"
```

This user can now use Single Sign-authentication. After the user logs into the Windows domain, the user no longer must enter username or password information during MS Outlook profile creation or login.

If Active Directory is unavailable at any point after setting up Single Sign-on, the Scalix Server prompts users for their regular domain password for authentication.

Non-SSO Kerberos Authentication

Non-SSO Kerberos authentication differs from SSO authentication in that the user is required to enter a password to log into MS Outlook, Scalix Web Access, or IMAP clients. However, the password the user enters is their Kerberos password instead of their Scalix password.

To configure non-SSO Kerberos authentication, do the following steps:

1 Make sure that Kerberos Server, Workstation, and Libraries are installed on a Scalix Server in your network. To verify that the required Kerberos rpms are installed on the system, enter:

```
rpm -qa | grep krb
```

You can obtain any missing Kerberos rpms from the Linux operating system installation CDs. See http://web.mit.edu/kerberos/www/ (Red Hat and Fedora) or http://www.pdc.kth.se/heimdal/ (SuSE) for more information.

2 Initialize the KDC. Enter:

```
omkrbinstall -r realm -s servername.domain -a username -p password
```

- `-r` specifies the realm that the KDC manages.
- `-s` specifies the name of the Scalix Server that hosts the KDC.
- `-a` specifies the principal fully qualified hostname of the KDC administrator.
- `-p` specifies the KDC administrator password.

The following prompt displays:

```
Checking MIT Kerberos installation...done. Initializing database '/
var//krb5kdc/principal' for realm 'REALM', master key name 'K/
M@REALM' You will be prompted for the database Master Password. It
is important that you NOT FORGET this password.

Enter KDC database master key:
```

3 Enter a password for the KDC database.

```
Reenter KDC database master key to verify:
```

4 Reenter the password for verification.

The following information displays:

```
Success! Kerberos database created, configured and started.
```

5 Create Scalix Service principal keytabs. Enter:

```
omaddprincs -s all -h server.domain -o filename.keytab
```

- Specifying all for the -s option creates one keytab file for all Scalix Services (Remote Execution Service, UAL, IMAP, and the Scalix Administration Console Service). However, you can also create individual keytabs for specific services. See the omaddprincs man page for more information.
- -h specifies the fully qualified domain hostname of the Scalix Server on which the Scalix Services are installed.
- -o specifies the keytab file name.

The following information displays if you use -s all to create a keytab:

Creating new Scalix principals in Kerberos database and keytab filename.keytab:

```
scalix-ual/server.domain@REALM
imap/server.domain@REALM
ubermanager/server.domain@REALM
res/server.domain@REALM
```

6 If necessary, manually copy the keytab file(s) to the Scalix Servers on which you want to use Kerberos authentication. See the *Scalix Installation Guide* for information about Scalix Administration Console-specific keytab deployment.

7 Modify the `~/sys/pam.d/`**ual.remote** file so that it appears as follows:

Note: Bold text indicates the lines that need to be modified.

```
# Standard Scalix Authentication
#
# Comment this out if you want to use one of the alternative
authentication
# schemes below.
# auth  required om_auth nullok

#
# Kerberos authentication 1
#
# With this scheme we attempt local authentication first and, if
that
# fails, we try Kerberos authentication. Note that if we do it the
other
# way around we run the risk of the KDC locking a principal account
for
# users that are known to both Kerberos and Scalix. See om_krb5(8)
for more
# information.
#

# auth  sufficient om_auth nullok
auth  sufficient om_krb5 use_first_pass
auth  required pam_deny

# Kerberos authentication 2
```

When you modify and save the ual.remote file, client sessions are initiated using UAL_INIT/UAL_SIGNON, and include the principal name and password. After the user is found (verified) in the USERLIST Directory, the authid value from their Directory entry is used to authenticate them through PAM (Pluggable Authentication Module). The om_krb5 module (with the authid value and password) is used to contact the KDC through the Kerberos client libraries.

8 For non-SSO Kerberos POP access, modify the ~/sys/pam.d/pop3 file so that it appears as follows:

Bold text indicates the lines that need to be modified.

```
#auth     required om_auth
account  required om_auth
password required om_auth
auth  sufficient om_krb5 use_first_pass
auth  required pam_deny
```

9 Scalix users will now authenticate against the KDC using their Kerberos password. If users experience problems while logging in to Scalix, make sure they are in the KDC.

```
kinit username
```

10 If the user is not in the system, enter:

```
kadmin.local

addprinc -pwd password username
```

11 This adds a user principle. To verify that the user was successfully added, enter:

```
listprincs
```

You should see a user principal for the user you created.

Make sure the user's authid value is set to username@DOMAIN.NET.

Using the Domain Password

If you want to use Kerberos authentication and have users use their Windows (Active Directory) domain password when logging into Scalix, complete all the steps in "Single Sign-on Kerberos Authentication", and then edit the ~/sys/pam.d/ual.remote file as described in Step 7 of "Non-SSO Kerberos Authentication".

Troubleshooting Kerberos and SSO

The following information describes how to troubleshoot problems with Kerberos and SSO.

SSO Troubleshooting

The UAL daemon log file (omsckd.log) can help you locate and resolve problems with Single Sign-on. This file records connections made to the UAL socket daemon (advmail.sckd) and the parameters used during the attempt to sign on.

To enable the omsckd.log file, do the following:

1. Kill the existing UAL socket daemon process:

   ```
   killall advmail.sckd
   ```

2. Enable the **omsckd.log** file:

   ```
   touch ~scalix/tmp/omsckd.log
   ```

3. Restart the UAL socket daemon process:

   ```
   /opt/scalix/bin/advmail.sckd
   ```

The **omsckd.log** file saves connection data after every attempted login. The output generated after an attempted SSO connection appears similar to the following:

```
connection from 10.11.108.233
*** Scalix UAL Session Start
13:45:51 Initial PhysicalBlockSize: 32
Initial LogicalBlockSize: 32
Initial LogicalBytesLeft: 0
Requested Flags: NO_LOW_LEVEL_ACKS LOGICAL_BLK_COMPRESSION
LOGICAL_BUFF_28_KB SERVER_PUSH_NOTIFS APPENDED_FT_DATA (0x2f00)
Requested Blocksize: 0
ActualFlags: NO_LOW_LEVEL_ACKS LOGICAL_BUFF_28_KB SERVER_PUSH_NOTIFS
APPENDED_FT_DATA (0x2d00)
Actual Blocksize:          28672
logical block: phys=33, logical=25, left=0, flags=0 (read so far: 0)
argv[0] = "99"
argv[1] = "0"
argv[2] = "0"
```

```
argv[3] = ""
argv[4] = "GSSAPI"
argv[5] = "imap"
argv[6] = "1"
logical block: phys=737, logical=729, left=0, flags=0 (read so far: 0)
GSSAPI (response): key
GSSAPI: accept_security_context
GSSAPI: gss_s_complete
GSSAPI (challenge): key
GSSAPI (response):
GSSAPI: send server parameters
GSSAPI (challenge): key
GSSAPI (response): key
GSSAPI: accept_client_parameters
GSSAPI: security mask: 0x1
GSSAPI: maximum token size: #ffffff
GSSAPI: authorization identity: "authid"
GSSAPI (challenge):
authentication completed successfully
```

The **omsckd.log** file provides information regarding any GSSAPI failure. GSSAPI is a generic API for client-server authentication that is implemented in all Kerberos 5 distributions.

For additional GSSAPI information and error messages, see GSSAPI documentation.

KDC Communication Troubleshooting

If you are having problems establishing communication to the KDC, you can use tcpdump to capture the data and then use tethereal to decode it for you.

1. To begin capturing the data, enter:

   ```
   tcpdump -s 0 -w /path/to/dump.file port 88
   ```

   This writes to a dump file for any communication where the source or destination port is 88 (the default KDC port).

2. Attempt an SSO login.

3. Kill the tcpdump process.

4. Execute `tethereal` against the dump file to decode the file:

   ```
   tethereal -r /path/to/dump.file
   ```

   This produces output that appears similar to the following:

   ```
   9  16.467979 192.168.123.123 -> 192.168.123.8 KRB5 AS-REQ

   10  16.469574 192.168.123.8 -> 192.168.123.123 KRB5 KRB Error:
   KRB5KDC_ERR_PREAUTH_REQUIRED

   11  16.470108 192.168.123.123 -> 192.168.123.8 KRB5 AS-REQ

   12  16.472615 192.168.123.8 -> 192.168.123.123 KRB5 KRB Error:
   KRB5KRB_ERR_RESPONSE_TOO_BIG

   13  16.472970 192.168.123.123 -> 192.168.123.8 KRB5 AS-REQ
   ```

```
14  16.474087 192.168.123.8 -> 192.168.123.123 KRB5 KRB Error:
KRB5KDC_ERR_PREAUTH_REQUIRED

15  16.474449 192.168.123.123 -> 192.168.123.8 KRB5 AS-REQ

16  16.476878 192.168.123.8 -> 192.168.123.123 KRB5 KRB Error:
KRB5KRB_ERR_RESPONSE_TOO_BIG
```

In the above output, the AS-REQ is followed by a pre-authentication error and then another attempted AS-REQ. The request then fails with ERR_RESPONSE_TOO_BIG (see “Packet Size Errors - Active Directory as the KDC” on page 243 for more information regarding this particular error).

5 If you run `tethereal` with the `-V` option, you can further decode the captured packets:

```
Frame 16 (171 bytes on wire, 171 bytes captured)

Arrival Time: Sep  6, 2004 17:43:52.528984000
Time delta from previous packet: 0.002429000 seconds
Time since reference or first frame: 16.476878000 seconds
Frame Number: 16
Packet Length: 171 bytes
Capture Length: 171 bytes

Ethernet II, Src: 00:0f:1f:9d:4a:c8, Dst: 00:0f:1f:6e:84:3c

Destination: 00:0f:1f:6e:84:3c (WwPcbaTe_6e:84:3c)
Source: 00:0f:1f:9d:4a:c8 (WwPcbaTe_9d:4a:c8)
Type: IP (0x0800)

Internet Protocol, Src Addr: 192.168.123.8 (192.168.123.8), Dst
Addr: 192.168.123.123 (192.168.123.123)

Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..0. = ECN-Capable Transport (ECT): 0
.... ...0 = ECN-CE: 0
Total Length: 157
Identification: 0x1e15 (7701)
Flags: 0x00
0... = Reserved bit: Not set
.0.. = Don't fragment: Not set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 128
Protocol: UDP (0x11)
Header checksum: 0xa466 (correct)
Source: 192.168.123.8 (192.168.123.8)
Destination: 192.168.123.123 (192.168.123.123)

User Datagram Protocol, Src Port: Kerberos (88), Dst Port: 32807
(32807)

Source port: Kerberos (88)
Destination port: 32807 (32807)
```

```
Length: 137
Checksum: 0xd71a (correct)

Kerberos KRB-ERROR

Pvno: 5
MSG Type: KRB-ERROR (30)
stime: 2004-09-06 16:44:39 (Z)
susec: 978618
error_code: KRB5KRB_ERR_RESPONSE_TOO_BIG (52)
Realm: COMPANY.ACME.NET
Server Name (Principal): scalix-ual platinum.company.acme.net
Name-type: Principal (1)
Name: scalix-ual
Name: platinum.company.acme.net
```

In the output above, you can see the Principal Names used to submit the requests.

Packet Size Errors - Active Directory as the KDC

There is a know issue relating to the use of UDP as a transport for Kerberos data when authenticating against an Active Directory KDC using the Red Hat 3.0 Kerberos client libraries.

During the authentication communications, the Active Directory server sometimes responds back to Scalix Server with KRB5KRB_ERR_RESPONSE_TOO_BIG. This happens when the data in a UDP packet is larger than the (arbitrary) limit.

Although Active Directory generates a service ticket when the client requests one, Active Directory embeds non-essential information relating to group memberships that causes the packet size to exceed the size limit.

To resolve the issue, you must upgrade Kerberos to version 1.3.1. The version installed with Red Hat 3.0 is version 1.2.7. Alternatively, you can make the following change in the Active Directory Server's HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc registry setting:

```
MaxDatagramReplySize 4000 (decimal) reg_dword
```

Chapter 20

Maintaining and Monitoring Scalix

This chapter describes the tasks you can perform to ensure that the Scalix messaging system operates properly. This chapter includes the following information:

Regular Administration Tasks

There are tasks that you should perform on a regular (if not daily) basis:

- "Verify Scalix Services"
- "Verify the Inbox of the Error Manager"
- "Verify that Messages Are Routing Through the Scalix System"
- "Check for Messages that Could Not Be Transmitted"
- "Check for System Errors"
- "Monitor Disk Usage"
- "Backing Up System Data"

Verify Scalix Services

All the relevant services must be running for mail to be delivered. Use the omstat command to regularly check the status of the services. You can also use the ommon command to report any services that have aborted since ommon was last run.

To turn the services on and off, see "Starting and Stopping Individual Services" on page 18.

Listing the Status of Services

You can list the status of Scalix Services using the command:

```
omstat -s
```

The status of each service displays. The status of a service can be one of the following:

Status	Description
Starting	Service has been started and is initializing.
Started	Service is running.
Stopping	Service has been stopped but is not completely shut down yet.
Stopped	Service is completely shut down.
Enabling	Service has been enabled and is initializing.
Enabled	Service is ready.
Disabling	Service has been disabled but is not completely shut down yet.
Disabled	Service is completely shut down.
Aborted	Service has been forced to shut down due to an error.
Part Aborted	Part of a service has been shut down due to an error.

Also shown is the time or date the service was last started or stopped, and the number of messages or users associated with the service.

The time is shown if the service was started or stopped today. Otherwise, the date is shown.

The Local and Remote Client Interfaces show the number of users currently using those services. The other services show the number of messages on the service queues.

Note	It is important to look for "aborted" services as opposed to "stopped" services. For example, enter: omstat -s | grep -i aborted

Listing the Status of Daemons

You can list the status of a daemon using the following command:

```
omstat -a
```

The status of a daemon can be one of the following:

Status	Meaning
Starting	Daemon has been started and is initializing.
Started	Daemon is running.
Stopping	Daemon has been stopped but is not completely shut down yet.
Stopped	Daemon is completely shut down.
Aborted	Daemon has been forced to shut down due to an error.
Part Aborted	Part of a daemon has been shut down due to an error.
NON-STOP	Daemon is configured to run continuously and is not affected by the `omon` and `omoff` commands. It can only be started and stopped by the Scalix system startup and shutdown commands `omrc` and `omshut` respectively.

Also shown is the time or date the daemon was last started or stopped. For daemons requiring subprocesses to work effectively, such as the Notification Server, the number of these processes running is recorded in the last column. 1 indicates one subprocess is running, 1 indicates that two subprocesses are running, and so on.

If a daemon has been stopped with a delay (using the omoff -d command), the status column indicates the delayed stop, and the date column shows the number of minutes remaining of the delay.

Note It is important to look for “aborted” daemons as opposed to “stopped” daemons. For example, enter: omstat -a | grep -i aborted

Verifying Scalix Queues

The Scalix message queues should be monitored frequently (hourly) to see if a build up of messages has occurred. Depending on the amount of traffic within your system, a threshold level could be used. For example, on a 1000 user Scalix Server, a good threshold might be if any of the queues reach 100 messages. The four standard queues to check are the Router, Internet, Local and Sendmail queues, each can be checked with the omstat utility.

For example, enter:

```
omstat -s | grep -Ei ‘router|local|internet|sendmail’
```

Listing Messages on a Queue

You can list messages on a queue using the command:

```
omstat -q queue [-m msg_ref]
```

Verify the Inbox of the Error Manager

The Inbox of the user configured as the Error Manager must be checked regularly for messages that cannot be delivered to local users and messages addressed to the Error Manager.

Verify that Messages Are Routing Through the Scalix System

If there is a problem with a service, messages will build up on the queue for that service and not be processed. Use the `omstat` command to regularly check the number of messages on each service queue. You can also use the `ommon` command to report any queues that have more than the configured number of messages on them.

Resubmitting Messages to the Service Router

You can resubmit messages to the Service Router using the command:

```
omresub -q queue [-i]
```

For details on how to use this command, see the online manual entry. For further information on resubmitting messages to the Service Router, see “The Service Router” on page 69.

If a User Does Not Receive a Message

- Check if the originator of the message received a Non-Delivery Report.

  The Non-Delivery Report can explain why the message was not delivered and where the problem occurred.

  The return of Non-Delivery Reports can be disallowed by the originator in some Scalix clients. If you suspect this has happened, ask the originator to resend the message and allow the return of Non-Delivery Reports.

- Check if the recipient has redirection or automatic actions set.

  Check the originator’s In Tray for a Non-Receipt Notification where a message has been auto-forwarded and then deleted, or deleted when it is unread.

- Check which interface or gateway the message used to enter the local Scalix system.

  If the originator is a local user of a client attached to Scalix, the message entered through the Client Interface. If a problem occurred in the Client Interface, the user would not have been able to create or send the message.

  If the originator was not a local user, then the user reporting the problem can know which interface or gateway the message used. If necessary, after completing this checklist, use the relevant checklist later in this section to check the interface or gateway.

- Check which interface or gateway the message used to leave the local Scalix system.

  Use the command `omshowrt -r` to see which queue the message would have been attached to. (Each queue name is associated with a specific service. See “Daemons” on page 18. for more information.

  If no queue is associated with the recipient address, the message will not be routed by this system, and a Non-Delivery Report will be returned to the originator (if allowed by the originator).

- Check the Scalix services are started.

  Use `omstat -s` to view the status of the services.

  The interface and gateway services identified in the previous steps, the Local Delivery service, and the Service Router should be started. If necessary, use `omon -s` *`service`* to start the services.

- Check the Event Log.

The Event Log will show if the message was received and delivered, and where the problem can be located.

The services must have been logging events at level 9 or higher when the message was sent. View the Event Log at level 9. If the logging level was less than 9, increase the level and ask the originator to resend the message.

Check for Messages that Could Not Be Transmitted

The Scalix interfaces and gateways are responsible for moving messages out of the local system and passing them onto a transport service or another mailing system. If you are using any of these interfaces or gateways, check the system to which the messages are being passed and ensure they are being processed correctly.

Use the omstat command to check for messages on the SMERR (Sendmail error) queue. Alternatively, use the ommon command to list all messages on the SMERR queue. If Sendmail encounters a routing problem with a message, the message is returned to Scalix and placed on the SMERR queue.

You can check for messages rejected by Sendmail using the command:

```
omstat -q SMERR [-m msg_ref]
```

If Messages Routed out Through the Sendmail Interface are not Received

- Check you can send a message to a Linux mail user on the destination machine using Sendmail directly. For example, if the node name of the destination machine is `saturn` and Sendmail is under `/usr/sbin`, type:

  ```
  /usr/sbin/sendmail root@saturn
  Test message, please phone me to confirm receipt Chris
  ```

  If you cannot send the message or the message is not received by the recipient, the problem is in Sendmail. See the Sendmail documentation for details.

- Check if the message is on the SMERR queue or ERROR queue.

  Use the `omstat` command in the form:

  ```
  omstat -q queue
  ```

  If the message is on the SMERR queue, then Sendmail did not accept the message. Use the `omstat -m` command to find out why Sendmail rejected the message. To resubmit the message to the Service Router, type:

  ```
  omresub -q SMERR -i.
  ```

  If the message is on the ERROR queue, it is likely the Service Router or the `xport.out` process has rejected the message. Try resubmitting the message. To do this, type:

  ```
  omresub -q ERROR -i
  ```

  If the message is rejected again, use `omqdump` to examine the message. Read the transaction file to find out why the message was not processed.

- Check the Sendmail mail queue for messages labelled "Deferred".

  The command to list the entries in the mail queue is usually `mailq`. See the Sendmail documentation for further details.

If Sendmail cannot send a message, the message is labelled as "Deferred". Sendmail tries to send the message every 30 minutes for the next 3 days.

Examine the message header files of the deferred mail (usually named qfmessage-id in the /var/spool/mqueue or /usr/spool/mqueue directory). The message header files specify why the message has been deferred.

You can check for deferred messages using the command:

```
omstat -d [-m msg_ref]
```

You can force a deferred message to be delivered early using the command:

```
omresub -d msg_ref
```

- Check the Sendmail log file.

  The log file shows if the message was received by Sendmail and sent on to the next system.

  The Sendmail log is usually in the file `/var/log/maillog`.

If Messages Routed in Through the Sendmail Interface are not Received

- Check the Sendmail mail queue for messages labelled "Deferred".

  The command to list the entries in the mail queue is usually `mailq`. See the Sendmail documentation for further details.

  If the `xport.in` process is not running, Sendmail is unable to deliver the message and it is labelled as "Deferred". Sendmail tries to send the message every 30 minutes for the next 3 days.

  You can check for deferred messages using the command:

  ```
  omstat -d [-m msg_ref]
  ```

  You can force a deferred message to be delivered early using the command:

  ```
  omresub -d msg_ref
  ```

- Check the Sendmail log file.

  The log file shows if the message was received by Sendmail and sent on to the next system.

  The Sendmail log is usually in the file `/var/log/maillog`.

- Check the Linux mailbox of the user `Scalix`.

  If there are messages in the Linux mailbox of the user `Scalix`, the problem is in the Sendmail configuration file. Messages addressed to the user `Scalix` are being delivered to the user `Scalix` rather than being processed by the `omxport` mailer.

- Check for messages in the `~/xport.errs` directory.

  Messages are placed in this directory if the xport.in process is unable to deserialize a message. (The `xport.errs` directory is created only when it is needed.)

Check for System Errors

System errors will likely become apparent during other checks you perform on a regular basis. However, to obtain the error report, use the ommon command. This will list all errors

and serious errors that have occurred since it was last run. In addition, these errors (and less serious errors) can be logged in the Event Log.

See "Configuring the Event Log Logging Levels" on page 256. for more information about the event log.

Monitor Disk Usage

Over time, the number of messages in the Message Store increases. To prevent users running out of disk space, monitor the disk usage using one of the standard Linux commands. In addition, the `ommon` command reports file system disk usage and the `omscan` command reports disk usage of each user (user mailbox size).

Encourage users to regularly delete old or unwanted messages. This keeps disk usage down to the minimum. Use the `omtidyu` command when you have to remove old or unwanted messages on behalf of individual users.

Check Disk Space Utilization

Always check available disk space on the volume where the Scalix message store resides. This check should occur quite frequently (hourly) and can occur using the Linux "df" command. Should a threshold be reached (%80-%90 of volume's space) an alert should be sent.

For example, enter:

```
df -t ext2
```

Check I-Node Utilization

Always check inode utilization on the volume where the Scalix message store resides as well. This check should occur quite frequently (hourly) and can occur using the Linux "df" command. Should a threshold be reached (%80-%90 of volume's space) an alert should be sent.

For example, enter:

```
df –i -t ext2
```

Removing Outdated Messages

One of the most typical tasks for any mail system is to set message retention limits for various containers, thus requiring a purging process on the outdated messages. The Scalix `omtidyallu` command can be used for this process.

For example, enter:

```
omtidyallu -w 30 -d 2>&1 >> $MAINTLOG
```

Backing Up System Data

As with any major enterprise messaging system, you should back up your data daily at minimum. Because Scalix sits on top of the Linux file system, the backup procedure is simply a matter of copying the file system. The backup should encompass all contents of the

instance's home directory (normally *var/opt/scalix* in a single-server or typical installation scenario), including all subdirectories and their files.

However, like most enterprise messaging systems, you must back up the appropriate files, and more importantly, when the files are not in use. This is because the Scalix database consists of files representing many different components (such as mailboxes, folders, messages, attachments, etc.) that dynamically reference each other. So files and pointers (reference information) are continually created and deleted during messaging transactions, making for a constantly changing environment. Therefore, any backup actions taken while the system is active can result in an incomplete and inconsistent copy.

There are two backup options:

- Shut down Scalix (omshut), and then copy, tar, tar-gz or rsync the contents of the instance's home directory to another location, then restart (omrc).
- Temporarily suspend write activity to Scalix (omsuspend), create a snapshot of the instance home directory, release the suspension, and then copy, tar, tar-gz or rsync the contents of the snapshot to another location.

Best Practices:

Some best practices for backups:

- Take advantage of snapshot capabilities. Logical Volume Manager, provided with Linux, provides snapshot functionality.
- After an initial backup, you can use an rsync procedure which backs up only the changes.
- When using the rsync method, use another Linux host if possible. This provides a redundant spare if needed, a message recovery server and multiple daily copies.
- Where applicable, use compression in the form of a tar -zxvf command.
- Always back up all of the instance home directory, including all subdirectories and their files.

Understanding the Synchronization Command

The general syntax when using rsync is:

```
#rsync -options source target
```

The "--delete" command is essential for use with Scalix, otherwise files deleted from the source are not deleted from the target, which can happen if mailboxes or messages are purposefully removed from your Scalix environment.

Alert	Do not mix up source and target options because switching the two around can cause critical problems.

Examples

- An example of the rsync command in a same-server backup is:

  ```
  #rsync -avz --delete /var/opt/ /backup
  ```

  This recursively copies all files from the directory */var/opt* to the directory */backup*. The files transfer in "archive" mode, which ensures that symbolic links, devices, attributes,

permissions, ownerships etc. are preserved. Additionally, compression is used to reduce the size of data portions. In this example the */backup* directory then contains a */scalix* directory (with all subdirectories), and perhaps a */jakarata-tomcat-5.0.2x* directory (with all subdirectories). This does mean the / or */backup* partition must have as much space available as is stored in the contents of */var/opt/,* which is typically rare. Run it initially, then run it again, notice the second time that only a few files copy. Write a message into a mailbox on Scalix, run it again, you'll notice now more files copy from */user* and */data* directories.

- An example of the rsync command when backing up to a different host

  ```
  #rsync -avz --delete /var/opt/ backup.company.local:backup
  ```

  This does the same as above, only it copies to the */backup* directory on the host "backup.company.local". The backup.company.local host is nfs mounted from the Scalix server. On the backup machine you can then set a nightly cron job to build a day-of-week .tgz file to another area, which is backed up to tape weekly.

Template Backup Scripts

Your Scalix software distribution contains a directory titled "admin_resource_kit", which contains a compilation of tools and scripts that are beneficial for administrators, including several template backup scripts. These scripts also are included in a larger tar/gzip file, which is typically prefixed with "sxbackup".

Using Logical Volume Manager

Scalix support tools such as the Knowedge Base and Support Forum provide technical notes and suggestions for proper configuration and use of Logical Volume Manager to enable snap-shot capabilities.

Verify the Log Files

Scalix includes log files that provide information about the state of the system. The following log files are located in the ~/logs directory.

Log Name	Description
audit	Logs information on Scalix services that can be analyzed for audit and statistical purposes. See <Xref_Color>Chapter , “Maintaining and Monitoring Scalix” on page 245 fo rmore information.
daemon.stderr	Logs standard error from the Scalix daemons; this usually contains details of any database errors that the daemons have encountered. This log is used as input to the ommon command. Each time this command is run, this log is copied by ommon and then deleted.
fatal	Logs errors and serious errors. These are also recorded in the Event Log. They are recorded here for scripts and programs that monitor the system, for example ommon. This log is used as input to the ommon command. Each time this command is run, this log is copied by ommon and then deleted.

Log Name	Description
ftlvis.log	Logs errors reported by the databases. This log is used as input to the ommon command. Each time this command is run, this log is copied by ommon and then deleted.
log.0,log.1,log.2	Log information about events occurring on the system. These logs are used to track events and identify problems with the system. See “The Event Log” on page 256 fo rmore information.

Periodic Administration Tasks

There are tasks that you should perform on a regular (if not daily) basis:

- “Verify and Repair Data Inconsistencies”
- “Add and Remove Local Users”
- “Maintain the Directory of Local and Remote Users”
- “Maintain the Public Distribution Lists”
- “Maintain the Routing Table”
- “Maintain the Public Folders”

Verify and Repair Data Inconsistencies

Use the omscan command to check for, and optionally repair, data inconsistencies.

If a process is aborted just before it deletes a file, an “orphan” can be created; or if a “child” file is deleted by mistake, the “parent” file will contain a reference to a nonexistent child file. The omscan service and omscan command can remove orphan files and references to children that no longer exist, as well as clearing out Scalix’s temporary storage areas.

You can run omscan in active or passive mode. In passive mode, it uses reports generated by the omscan Server to identify data inconsistencies. In active mode, it scans the data directly.

You can configure the omscan Server by editing the file ~/sys/omscan.cfg. In addition, you can use the omsetsvc command to specify how you want the Server to run.

Running omscan in active mode can consume considerable resources, and is not generally recommended.

Note	The omscan Server (and the omscan command running in active mode) makes some checks on mounted volumes but, to avoid severe consequences, do not allow disks containing anything under the instance home directory (usually /var/opt/scalix/) to be unmounted while the omscan Server or the omscan command is running.

If Database Problems are Reported

1 Check the database monitor daemon is running.

2 Look for the daemon omdbmon in the process listing. If the daemon is not running, start it. As root type:

```
/opt/scalix/bin/omdbmon
```

3 Check the Event Log for database problems.

4 If there are database errors, use omsolve to find out more about the errors.

5 Check the ~/logs/ftlvis.log file.

6 If there are errors, use omsolve to find out more about the errors.

7 Check the consistency of the database.

8 Use the dbcheck utility. If dbcheck identifies problem files with .dat extensions, then data is missing. Reload the database from a previous system backup.

9 If dbcheck identifies problem files with .key extensions, then try rebuilding the database key files using the command omdiropt.

Scan the Scalix message store for corruption

The Scalix message store should be checked for any corruption on a weekly basis. Disk defragmentation is not required with Linux.

Example:

```
omscan -a -f -x 2>&1 > $(omrealpath '~/logs/omscan.out')
```

Add and Remove Local Users

Over time, users will need to be added to and removed from the system. To add or remove users, see "Adding a Local User" on page 8.

Maintain the Directory of Local and Remote Users

Addressing users can be made easier if their addresses are included in a Directory. For a Directory to be effective, it must be kept up to date as users throughout the network are added and removed.

As users are added to and deleted from a Directory, the Directory structure becomes fragmented making searches of the Directory slower. To rebuild the Directory use the omdiropt command from time to time.

Maintain the Public Distribution Lists

Changes in the Scalix user population throughout the network must be reflected in any local Public Distribution Lists that refer to those users. Making these changes can be part of the processes you have established with other Administrators for maintaining Directories, in which Public Distribution Lists are held. If not, the first signs that a user referred to in a Public Distribution List no longer exists or has changed address will be Non-Delivery Reports in the In Tray of the Error Manager if the user is on your system, or Non-Delivery Reports returned to the user using the Public Distribution List.

Maintain the Routing Table

The Routing Table determines which system a message must be passed to for a specific address. As addresses are added and removed from the network, so your Routing Table might need to be modified to reflect those changes. You need to establish a procedure with other Administrators for keeping it up to date.

Maintain the Public Folders

Using a client that can access Public Folders, periodically check any Public Folders on the system. Delete any Public Folders or items that are not being read anymore. Check that each Public Folder has appropriate items on it.

To automatically delete items under top-level Public Folders that are older than a specified number of days, use the ommaintbb command.

The Event Log

All Scalix services can log information to the Event Log. The level of detail logged by a specific service (the "logging level") is controlled by the omconflvl command. The omshowlvl command shows the current logging level. The command omshowlog is used to view the Event Log.

The Event Log is a circular log. As the log becomes full, earlier information is overwritten by later information. The size of the Event Log is configured using omconflvl. The default log size is 300 kilobytes. As the logging level is increased, more information is written to the log. To maintain a log spanning the same time period, the size of the log must be increased when the logging level is increased.

The Event Log is held in up to three binary files, log.0, log.1, and log.2 in the directory ~/logs. Each file can grow to be approximately one-third of the configured log size. As log.0 becomes full, log.2 is deleted, log.1 is renamed to log.2, log.0 is renamed to log.1, and a new log.0 is created.

A "lock" file (~/logs/lock_log) is used to prevent two or more processes writing to the log simultaneously.

Configuring the Event Log Logging Levels

You can configure the Event Log logging levels using the command:

```
omconflvl service level
```

The following table lists event logging levels.

Level	Meaning
1	Log SERIOUS ERRORS that prevent the process from continuing.
3	Log ERRORS where the process can continue, and SERIOUS ERRORS.
5	Log WARNINGS of possible problems, ERRORS, and SERIOUS ERRORS.

Level	Meaning
7	Log REPORTS of successful execution of commands, WARNINGS, ERRORS, and SERIOUS ERRORS.
9	Log REPORTS from the standard mailing processes, other REPORTS, WARNINGS, ERRORS, and SERIOUS ERRORS.

Configuring the Event Log Size

You can configure the size of the Event Log using the command:

```
omconflvl -s log-size
```

Displaying the Event Log

You can display the Event Log using the command omshowlog.

Scalix Application Logs

Scalix Applications, such as the Administration Console, the Remote Execution Service, and Scalix Web Access, generate log files that report activity and list events that occur when these applications are in use.

The Scalix Administration Console generates log information in the TOMCAT_HOME/logs/caa.log file. The Remote Execution Service outputs log information to the TOMCAT_HOME/logs/res.log file

You can set the logging level for the Scalix Administration Console in this file:

```
/etc/opt/scalix/caa/config/log4j.properties.
```

The logging level for the Remote Execution Service can be set in this file:

```
/etc/opt/scalix/res/config/log4j.properties.
```

You can change the type of information generated in the caa.log file by modifying the log4j.rootLogger and log4j.logger.com.scalix.caa.util.CAALogger parameters in the Administration Console's log4j.properties file with one of the following logging options:

INFO (default) - logs only WARN, ERROR, and FATAL events.

DEBUG - all activity and events are logged.

REPORT - logs usage and login information.

Similarly, you can change the type of information generated in the res.log file by modifying the log4j.rootLogger and log4j.logger.com.scalix.sac.rest.util.RESLogger parameters in the Remote Execution Service's log4j.properties file with one of the logging options listed above.

By default, The maximum size of the caa.log file is 100 KB. When that limit is reached, all log information is removed and the file is reset. Typically, the caa.log file grows at a much faster rate when you set logging to DEBUG. The default setting in the log4j.properties. files for both the Administration Console and the Remote Execution Service is INFO.

You can modify the maximum size of the rolling caa.log and res.log file. Modify the log4j.appender.rfile.MaxFileSize parameter in each component's respective log4j.properties file with the desired file size.

Scalix Web Access generate log information to the TOMCAT_HOME/logs/scalix-swa-date.log file. A new log file is created every day (assuming activity occurs in Scalix Web Access).

Scalix Connect Support Tab Logging Options

The server UAL Trace level can also be set on the Support tab of the Scalix Connect Properties window.

Note	Scalix Connect for Microsoft Outlook can be used only by Premium users. For more information, see "About Scalix Product Editions".

1. Navigate to Start > Control Panel > Mail > Services, select **Scalix Service (with server store)** and click **Properties**.
2. Click the **Support** Tab in the Scalix Properties window.
3. You can specify the location of the log file. The default location and filename is C:**scalixmail.log**. This file is overwritten every time you log in to the Scalix Server.

Logging errors and events can create large files and degrade system performance. Scalix Corporation recommends only temporarily using logging options other than "Errors Only" to troubleshoot problems with Scalix Connect.

You can also specify the information that is logged in the scalixmail.log file for the following providers:

- Transport
- Address Book
- Message Store

The following lists the information you can log:

- Errors only (the default)

Logs only error messages.

- MAPI entry and exit points

Logs operations involving Scalix MAPI extensions, for example, Scalix AutoForward/AutoReply.

- MAPI entry and exit points + commentary

This option gives additional logging information such as Scalix properties and values used.

Note that "Errors only" is selected by default.

There are two additional checkboxes you can use to enable the logging of internal functions and the logging of UAL command/reply data (including the trace level). Use these options if problems occur between the Outlook client and the Scalix Server.

To further help you resolve problems, you can increase the Server Trace Level to enable Scalix Connect to log more data in the scalixmail.log file.

Additional Log Files

The ommapi.log file is a permanent diagnostic log file to which information is added during each Scalix Connect "session" (\WINNT\ommapi.log). The file is automatically truncated when it reaches a certain size and contains diagnostic information similar to that in the scalixmail.log file.

The omname.log file logs name parsing problems (performed by omname32.dll) and is enabled in the \WINNT\ual.ini file as follows:

```
[Name Parsing]
Logfile=\OMNAME.LOG
LogFlags=1
```

The omname.log file is not created unless LogFlags is set to 1.

Running the Scalix Monitor Program

You can run the Scalix monitor program using the command:

```
ommon -m mount_point -p percentage -q queue_limit -u
internet_mail_address
```

The ommon utility is normally scheduled (using the Linux CRON command) to run twice a day. For example, to run ommon every day at 7 AM and 12 AM (with any error messages redirected to a temporary file), put the following entry in the crontab file of root:

```
0 7,12 * * * /opt/scalix/bin/ommon -q5 -p80 > tmp/ommon
```

Managing Users

The following information describes how to manage mailboxes and send and receive capabilities for Scalix users.

Controlling a User's Ability to Send and Receive Mail

There are a number of mechanisms by which you can put limits on a user's ability to send and receive mail.

- You can modify individual users to prevent them receiving any messages.
- You can create rules to control receipt and delivery based on user service levels that you define.
- You can use sanctions to automatically prevent users from receiving or sending mail when they exceed their message store limits.

Remember to restart the Service Router after you modify ruleset files.

Preventing Individual Users from Receiving Messages

To close a user's mailbox to incoming messages, use the ommodu command with the -r option.

For example, enter:

```
ommodu -o "Chris Wolf/ny,hq,mis" -r N
```

This prevents the specified user from receiving any mail.

When a user whose mailbox has been closed signs on to Scalix, a message in their Inbox will inform them that mail receipt is disabled. When a user attempts to send a message to a user whose mailbox is closed, they will receive a Non-Delivery Notification.

Using this option has no effect on the user's ability to send mail.

To allow a previously disallowed user to receive mail, issue the following command:

```
ommodu -o "Chris Wolf/ny,hq,mis" -r Y
```

Checking Mailbox Status

To check the status of an individual user, use the omshowu command with the -n option to specify the user. For example,

```
omshowu -n "Chris Wolf/ny,hq,mis" -r
```

To list all users who have closed mailboxes, issue the following command:

```
omshowu -r N
```

To list all users who have open mailboxes, issue the following command:

```
omshowu -r Y
```

Service Levels

A service level is a number that a Scalix administrator assigns to a user. The administrator can then create rules that control mail receipt and delivery for classes of users based on their service level.

To assign a service level to a user, use the ommodu command with the -s option.

For example, enter:

```
ommodu -o "Chris Wolf/ny,hq,mis" -s 100
```

This assigns a service level of 100 to the user Chris Wolf.

To make use of service levels in determining receipt and delivery policy, you must create mail delivery rules that utilize them.

Service levels have no significance other than their use in mail delivery rules. However, you should reserve a value of 0 to mean that no service level is configured.

See "Scalix Interfaces and Gateways" on page 25 for more information about rules.

Displaying Service Levels

Use the omshowu command to display a user's service level.

For example, enter:

```
omshowu -n "Chris Wolf/ny,hq,mis"
```

To display all users with a specified service level, use one of the following commands:

```
omshowu -v eqN
```

This displays all users with a service level of *N*.

```
omshowu -v gtN
```

This displays all users with a service level greater than *N*.

```
omshowu -v ltN
```

This displays all users with a service level less than *N*.

```
omshowu -v neN
```

This displays all users with a service level that is not equal to *N*.

Controlling Receipt and Delivery Using Service Levels

You can control receipt and delivery of messages based on the service level of the recipient or sender by creating rules that use the RECIPIENT-SERVICE-LEVEL or the SENDER-SERVICE-LEVEL message filter attribute.

Examples

You could create the following rule:

```
RECIPIENT-SERVICE-LEVEL=100 ACTION=REJECT
```

This would prevent the message from being delivered to any recipient in the distribution list who has a service level of 100, and cause a Non-Delivery Notification to be returned to the sender. The intended recipient does not receive a notification when this occurs.

Instead of preventing delivery outright, you can use a rule to defer delivery until outside office hours. For example,

```
RECIPIENT-SERVICE-LEVEL=50 ACTION=DEFER TIME=09:00-18:00
```

This causes recipients whose service level is 50 to have receipt of all mail addressed to them deferred between 09:00 and 18:00.

An example rule to prevent delivery of messages is:

```
SENDER-SERVICE-LEVEL=10 ACTION=DISCARD
```

This causes any message sent by a user with a service level of 10 to be discarded.

Exceptions

- Non-Delivery Notifications are always delivered, even if receipt is disabled.
- The behavior of Scalix when receipt or delivery is restricted can be modified by the mail clients in use. For example, some clients can not return to the sender a message that was sent to a closed mailbox.

Message Store Size Limits

To prevent the Scalix message store from growing too large, you can configure limits on the message store components as follows:

- Inbox
- Filing System (folders)
- Deleted Items
- Outbox
- Distribution List Area
- The overall message store (comprising the total space required for all the above components)

For each of the above components, you can set a default value that applies to all users, and you can override those defaults for selected users by setting user-specific values.

Once you configure limits, you must then configure the sanctions to be applied to those users who exceed them. If you do not configure sanctions, users can exceed their limits without effect.

"Configuring Sanctions for Users Who Exceed Their Limits" on page 263 describes the sanctions you can configure for those users who exceed their limits.

Setting System-Wide User Defaults

Use the omlimit command to set system-wide message store limits, as follows (N is the size limit in Kbs):

- Default Inbox limit

  ```
  omlimit -i N -g
  ```

- Default Filing System limit

  ```
  omlimit -f N -g
  ```

- Default Deleted Items Deleted Items limit

  ```
  omlimit -w N -g
  ```

- Default Outbox limit

  ```
  omlimit -p N -g
  ```

- Default Distribution List Area limit

  ```
  omlimit -d N -g
  ```

- Default overall message store limit

  ```
  omlimit -g -m N
  ```

Overriding Default Limits for Specific Users

To set user-specific message store size limits, use the above commands but include the `-u` option to specify the user whose limits you are setting and omit the `-g` option. For example, to set limits for the user Chris Wolf at ny,hq,mis, issue the following commands (*N* is the size limit in Kbs):

- Default Inbox limit

  ```
  omlimit -u "Chris Wolf/ny,hq,mis" -i N
  ```

- Default Filing System limit

  ```
  omlimit -u "Chris Wolf/ny,hq,mis" -f N
  ```

- Default Deleted Items limit

  ```
  omlimit -u "Chris Wolf/ny,hq,mis" -w N
  ```

- Default Outbox limit

  ```
  omlimit -u "Chris Wolf/ny,hq,mis" -p N
  ```

- Default Distribution List Area limit

  ```
  omlimit -u "Chris Wolf/ny,hq,mis" -d N
  ```

- Default overall message store limit

  ```
  omlimit -u "Chris Wolf/ny,hq,mis" -m N
  ```

Configuring Sanctions for Users Who Exceed Their Limits

You can configure sanctions that will be applied automatically to any user who exceeds one of the limits that are configured.

You configure sanctions using the omlimit command with the -e option. To configure sanctions for a specific user, include the -u option. To configure default sanctions for all users, include the -g option.

All sanctions, apart from preventing a user from receiving mail, apply when any of the configured limits is exceeded.

The list of possible sanctions, and the commands used to configure them, are as follows:

- Prevent a user from receiving mail

  ```
  omlimit -g -e r
  omlimit -u "Chris Wolf/ny,hq,mis" -e r
  ```

  This sanction only applies when the user exceeds the overall message store limit. If you have not configured an overall message store limit (using `omlimit -m N` as described above), this sanction will have no effect.

- Disable or restrict a user's ability to send mail

  ```
  omlimit -g -e s
  omlimit -u "Chris Wolf/ny,hq,mis" -e s
  ```

  This command does not directly affect a user's ability to send or receive messages. This command enables you to create message delivery rules that specify the sanctions to apply to the user. See "Creating Rules to Implement the "Disallow Sending" Sanction" for more information. For some non-Scalix intranet recipients, you can bypass these sanctions by sending messages using SMTP. This is the mechanism typically used by POP3 or IMAP4 clients.

- Cause a message to be sent to a user warning that they are over the configured limit

  ```
  omlimit -g -e u
  omlimit -u "Chris Wolf/ny,hq,mis" -e u
  ```

If required, Scalix sends warning messages to a user's Inbox when the user logs into their mailbox. Also, Scalix sends a warning message if a user performs an action in their mailbox (such as filing messages) that causes the mailbox to exceed the limit you specify. Warnings are not resent if the user received the warning within the last 1 day. You can change the default value of 1 day by configuring the *OMLIMIT_MIN_WARN_INTERVAL* parameter in the **general.cfg** file.

In addition, if an action the user takes whilst accessing their mailbox takes the mailbox over a limit, then a message will be added to reflect that fact.

- Cause an error message to be generated when a user attempts to create an item in the Filing System or Distribution List Area.

```
omlimit -g -e e
omlimit -u "Chris Wolf/ny,hq,mis" -e e
```

  This sanction does not apply to any overall message store limit that is set.

- Cause a message to be sent to the Error Notification User telling them that a user is over the configured limit

```
omlimit -g -e a
omlimit -u "Chris Wolf/ny,hq,mis" -e a
```

Creating Rules to Implement the "Disallow Sending" Sanction

This section describes how to create a message delivery rule to prevent or restrict the ability of a user to send mail when they exceed their configured message store limit.

Users using POP3 or IMAP4 clients send messages using SMTP. Sending sanctions only apply to these messages if the messages traverse the service router on the system that contains the mailbox of the sender. This is typical for messages sent to other Scalix intranet users. However, messages sent to the public internet traverse the Sendmail interface and therefore bypass the sending sanctions.

Before you create a rule to implement this sanction, you must issue the omlimit command with the -e s option. Otherwise any relevant rules that you configure will not operate.

For example, to restrict the ability of user Chris Wolf to send mail when he exceeds any of the limits configured on his message store, you must first issue the following command:

```
omlimit -u "Chris Wolf/ny,hq,mis" -e s
```

Then create a rule as described below.

Create a rule to implement this sanction using the `OMLIMIT-EXCEEDED` message filter attribute.

For example, to prevent a user from sending messages when they exceed one of their configured limits by any amount, create the following rule:

```
OMLIMIT-EXCEEDED=100 ACTION=REJECT
```

To prevent users from sending mail if they are at 110 percent or more of a limit, and generate a warning for users who are between 90 and 110 percent of a limit, create the following rules:

```
OMLIMIT-EXCEEDED=110 ACTION=REJECT
OMLIMIT-EXCEEDED=90 ACTION=ALLOW NOTIFY="warning message"
OMLIMIT-EXCEEDED=100 ACTION=RETURN NDN-INFO="warning message"
```

Monitoring the Message Store

You can display for each user which limits are configured and which sanctions are implemented using the omlimit -r command. For example,

```
omlimit -u "Chris Wolf" -r
```

```
Name                          : Chris Wolf /lim,test
Message Store Size Limit      : 25Kb
Inbox Size Limit              : 10Kb
Filing System Size Limit      : 10Kb
Deleted Items Size Limit      : 5Kb
Outbox Size Limit             : 5Kb
Distribution List Area Limit  : 2Kb
Omlimit Sanctions Enabled
(u)                           User receives a message warning of the
                              over limit
(e)                           UAL errors are reported to the client when
                              attempting to move items to full areas.
(s)                           Messages sent are subject to any delivery
                              rules that utilize the OMLIMIT-EXCEEDED
                              message filter attribute
```

You can list all users who are using a specific percentage of their configured limits. Scalix also lists the area(s) that match this criteria for a user. For example, to list all users using 90% or greater of their configured limits, enter:

```
omlimit -g -x 90
```

```
Users using at least 90 % of their allocated size limits
Name                          : Marian Brand/lim,test/CN=Marian Brand
Inbox, Total Message Store
Name                          : Chris Wolf /lim,test/CN=Chris Wolf
Distribution List Folder
Name                          : Julie Wills /lim,test/CN=Julie Wills
Inbox
```

Generating a Report

You can use the omscan command to report on configured limits for the message store. The generated listing includes configured limits and actual usage for each message store component.

For example, use the following command to generate a report for all Scalix users on the server:

```
omscan -A -u -o 3
```

Archiving Messages

The Archiver service allows you to archive all messages that traverse the Scalix Server. You can then view the archived messages using a third-party viewer.

By default the Archiver service starts when you start a Scalix instance, but does not archive messages until you enable the Archiver in the **general.cfg** file. After you enable and configure the Archiver options in the **general.cfg** file, you must also restart the Archiver and the Service Router as follows:

```
omoff -d0 sr archiver
omon sr archiver
```

The Archiver copies all messages that route through the Service Router and saves them to a location you specify in the **general.cfg** file. If required, the Archiver can have auxiliary processes (created with `omsetsvc -x`).

UAL Client Interface Tracing

The UAL protocol offers a number trace options to record the interaction between a UAL client and the Scalix Server:

Bit	Decimal Value	Description
Bit 1	1	Raw, unformatted, command/reply trace without file transfer information.
Bit 2	2	Command statistics
Bit 3	4	Message Store file names.
Bit 4	8	Full formatted command/reply trace with file transfer information.
Bit 5	16	Raw, unformatted, command/reply trace with file transfer information.
N/A	N/A	Server procedure trace.

A trace is generally activated by a client. The method of turning on a trace varies from client to client: refer to the relevant client documentation for details. However, because a trace can generate a large volume of data and significantly affect system performance, it is also possible to set the UAL trace level on a system-wide basis, with the option UAL_FORCE_TRACE_LEVEL in the general configuration file. This is useful in limiting the disk space used by trace files.

Command/Reply Trace

The command/reply trace records the commands and replies exchanged between the UAL client and the Scalix Server. Some options also include information about file transfers.

Output from the different command/reply traces is saved on the server in the ~/tmp directory with a file name derived from the Scalix ID number of the client user:

- OM*user-no*N.trc: raw, unformatted command/reply trace
- OM*user-no*.stats: command/reply statistics trace
- OM*user-no*U.log: full formatted command/reply trace with file transfer information

You can configure separate trace files for concurrent sessions by using the UAL_TRACE_FILE option in the ~/sys/user.cfg file.

When the full formatted command/reply trace option is used, files transferred between the client and server are also saved in the ~/tmp directory with a suffix of .fnnnn, where nnnn is a number starting at 001 (OMuser-noU.fnnnn).

Procedure Trace

The procedure trace records the execution of the server process handling a particular UAL client connection.

Different levels of procedure trace can be recorded. The levels are equivalent to the Event Log logging levels multiplied by 100. For example, to produce a detailed procedure trace, use the Event Log logging level 13 multiplied by 100; that is 1300.

Output from the procedure trace is written to the Event Log and can be examined using omshowlog.

Message Store File Name Trace

Use this trace to locate the constituent files of a message in the Message Store.

When turned on, the file names of each part of a message in the Message Store are displayed in the Subject field of the user's client. File names are prefixed by a tilde (~) representing the instance's home directory (usually /var/opt/scalix on a typical single server installation).

IMAP4 Server Process Tracing

You can trace information from the in.imap41d process by setting the IMAP_LOGLEVEL option and/or the IMAP_LOGFILE and IMAP_UAL_TRACE_LEVEL options in the general configuration file (~/sys/general.cfg), or in the appropriate client-specific or user-specific file. For details of these options, see "Configuration Options" on page 301.

For each instance of in.imap41d, a log file is created. The name and location of this file is determined by the value of the IMAP_LOGFILE option.

Because the in.imap41d process is a Scalix remote client communicating with the Scalix Server via the UAL protocol, you can also use the standard UAL tracing options to see what the UAL has done to service the IMAP4 session.

Testing the LDAP Server

The LDAP Server daemon generally starts when Scalix starts. If you have turned the daemon off (using the omoff command), you can explicitly start the LDAP Server with the following command:

```
omon -a slapd
```

If the LDAP Server has been correctly configured, the daemon starts, puts itself in the background, detaches from its controlling terminal, and listens for connection requests.

If the LDAP Server does not start, you can debug it by having the daemon print diagnostic information on the terminal on which it was started rather than putting itself in the background. You can obtain a basic level of debugging by entering the following command:

```
/opt/scalix/bin/omslapd -f ~scalix/sys/slapd.conf -d13
```

This records the LDAP requests as well as the arguments passed to the search request and the corresponding Scalix search arguments. The Scalix arguments and results contain non-printing characters and, occasionally, very long lines.

If you suspect a problem with the LDAP protocol, you can obtain a higher level of debugging by entering the following command line:

```
/opt/scalix/bin/omslapd -f ~scalix/sys/slapd.conf -d31
```

This provides additional information, including the contents of the Protocol Data Units (PDUs) received from and sent to LDAP clients. The debugging levels are the same as those that can be set in the slapd.conf configuration file.

POP3 Server Process Tracing

You can trace information from the POP3 Server process by setting the UAL_POP3_TRACE option in the ~/sys/general.cfg file.

For each instance of the POP3 Server, a file is created under ~/tmp. The file is called pop.PID.client_hostname.UID, where PID is the process ID of the POP3 Server process, client_hostname is the host name of the client machine (for example localhost, or host1.sales.org.com), and UID is a unique identifier.

Updating Scalix After Changing the Name of the Server

When you change the Fully Qualified Domain Name of the system on which a Scalix Server operates (/etc/hosts file), you must execute the sxmodfqdn command to ensure that Scalix applications (such as the Scalix Administration Console) continue to operate properly.

The sxmodfqdn command executes a script that updates SYSTEM directory entries with the new Fully Qualified Domain Name of the Scalix Server, and updates various other Scalix items with the new hostname.

You must manually update the following items with the new Fully Qualified Domain Name of the system on which Scalix operates:

- `/etc/opt/scalix/webmail/`**swa.properties** (Scalix Web Access)
- `~/sys/`**smtpd.cfg**
- `/etc/mail/`**sendmail.cf**
- The `INET_DOMAIN_NAME` parameter in the `~/sys/`**general.cfg** file.
- Any mail address rules (if applicable)

After you change the hostname, execute the `sxmodfqdn` command, and modify the items listed above, shutdown the Scalix Server (`omshut`), and reboot the system.

After you login to the system, start Tomcat (if necessary):

```
cd /TOMCAT_HOME/bin
./startup.sh
```

Automating Maintenance and Monitoring Tasks

Scalix provides a sample script named `ommaint` in the `/INTEGRATION` directory on the Scalix Installation CD. `ommaint` executes the major maintenance and monitoring activities for the Scalix application, and uses e-mail to report the results of each task and event.

For more information, see the comments in the `ommaint` script file.

Chapter 21

Non-Delivery Notification

When a Message Transfer Agent (MTA) fails to deliver a message to the intended recipient, the MTA generates and sends a non-delivery report. This chapter includes the following information:

Non-delivery Notification Overview

When a message cannot be routed to or delivered to the recipient's Message Store, Scalix services such as the Service Router and the Local Delivery Service perform a non-delivery notification service. That is, they generate a non-delivery report and return it, along with the original message (by default), to the message originator and to the Error Manager for the Scalix system that failed to deliver the message.

- The non-delivery report is a text message. It is returned to the originator as a transaction file record that is expanded into a readable form at the receiving Client Interface or gateway. The report gives the following information:
- The O/R Address of the originator.
- The recipient to whom the message could not be delivered.
- The reason for the failure.

Reasons for Non-delivery Notification

The following table shows the reasons non-delivery reports are generated within an Scalix system, listing the Scalix service detecting the error, and providing a cross-reference to additional information.

Reason	Service
Address can not be routed or resolved	Service Router

Reason	Service
Message looping around system	Service Router
	Local Delivery Service
Recipient can't be found in local user list	Local Delivery Service
Message fails to pass through a gateway	relevant gateway
The latest delivery time expired before message could be delivered	Local Delivery Service
Access to a service denied because originator does not have sufficient access capability	relevant service
PDL expansion disabled	Local Delivery Service
Message contains virus infected file	Service Router
Mailbox size limit has been exceeded	Service Router
Service level for mailbox does not allow operation	Service Router
Mailbox currently exceeding its size limit	Local Delivery Service
Mailbox disabled by administrator	Local Delivery Service

Notification by External Systems

When a message that leaves the Scalix system cannot be delivered to the external recipient, it is up to the external mail system whether or not to return a non-delivery report.

If an external mail system does generate a non-delivery report, it is returned to the originator. However, unlike an Scalix non-delivery report, a copy is not sent to the Error Manager on the incoming Scalix gateway or interface system.

When the Error Manager Does Not Receive Reports

The Error Manager (for the Scalix system that failed to deliver the message) does not receive the non-delivery report and original message in the following cases:

- No Error Manager has been configured for the server.
- The original message is marked as "private".
- The intended recipient is on the blind carbon copy (BCC) part of the distribution list.
- The non-delivery report has been generated for a simple address problem and the NDN_EM_SERIOUS_ONLY option is set in the general.cfg file.

For details on the Error Manager, see “Error Manager” on page 297. For details on the general.cfg file, see “Configuration Options” on page 301.

Chapter 22

Audit Log

The Audit Log is one or more text files containing information that you can analyze for audit and statistical purposes.

The chapter includes the following information:

Overview

By default, the Audit Log is a single file named ~/logs/audit. You can use standard Linux text-manipulation utilities to process the file that constitute the "Audit Log" and create customized reports.

Over a period of time, the Audit Log increases in size and can become very large. Scalix recommends implementing a method to control the size of the Audit Log. For example, you can create a script that renames the Audit Log files with a name that reflects the time the script is executed. You can use a cron process to execute the script on a daily basis and enable the script to overwrite the Audit Log files that were generated the previous day. A sample of such a script is in the /opt/scalix/examples/general directory.

Audit Logging Levels

A logging level determines the level of detail written to the Audit Log by a Scalix service.

Logging levels can be any number from 0 (zero) upwards, and are set for specified services with the omconfaud command. You can display the logging levels using the omshowaud command. The lower the logging level set for a service, the less information is logged for that service; the higher the level, the more information logged.

The following table lists the range of logging levels (and the type of information that is logged) typically used for the Audit Log.

Level	Logging Information
0	Do not log anything. Turn off logging for the specified service.
1 . . .	Log the time at which an event occurred.
5 . . .	Log general message and user information.
11	Log part-by-part information about messages and disk usage by container.

Complete information about the data logged for a service is specified in the Audit Configuration File (~/sys/audit.cfg).

The audit.cfg File

The Audit Configuration File (~/sys/audit.cfg) specifies:

- what operations are logged (there can be more than one operation logged for a Service)
- at what level they are logged
- to what file or files they are logged

The file or files to which information is written is referred to collectively as the Audit Log.

An excerpt from the Audit Log Configuration file is shown below:

```
# service router
% 1 routing                    ~/logs/audit
1 time                         1
2 type                         5
3 ua-message-id                1
4 ua-ack-id                    3
5 mta-message-id               1
6 mta-ack-id                   3
10 subject                     11
20 sensitivity                 7
21 priority                    7
22 importance                  7
23 created-locally             7
30 originator                  9
35 designate-originator        9
40 part-type                   11
41 part-size                   11
42 message-size                5
43 part-count                  11
45 hop-count                   7
50 recipient-from              9
51 recipient-to                9
52 recipient-cc                9
53 recipient-bcc               9
60 ack-req                     7
61 message-filter-info         9
70 queue                       9
80 delivered-count             7
91 orig-recip                  7
92 new-recip                   7
95 non-delivery-reason         7
96 all-recips-non-deliv        7
98 max-nest-depth              11
```

The first two lines indicate that this excerpt details the routing operation of the Service Router, and that the file to which data is written is ~/logs/audit. This is the default Audit Log.

Grouped under the operation indicator (1 routing) is a list of field names showing the type of information that can be logged. On the right of each field name is a number; this represents the logging level at which that field is logged. Therefore, in the excerpt above, logging of information associated with 21 priority (the priority of messages) only takes place when the logging level for the Service Router is set (with the omconfaud command) to a level of 7 or higher.

For example, to log only the size of messages passing through the Service Router, and to record the information in a file called /tmp/msg-size, do the following steps in the audit.cfg file:

1. Ensure the number on the right of the field name is *greater* than 1 for all field names in the routing operation.
2. Make the number on the right of the field name `message-size` *equal* to 1.

3 Change the file name ~logs/**audit** to /tmp/**msg-size**.

4 Set the logging level for the Service Router to 1 by executing the following command:

```
omconfaud router 1
```

5 Restart the Service Router by entering:

```
omoff -d0 -s sr
omon -s sr
```

Audit Log File Format

The Audit Log is one or more text files containing groups of lines. Each group of lines is separated by one or more blank lines. Each group begins with an operation indicator, which shows what operation caused the following lines to be logged. Following the operation indicator, other lines in the group begin with a field name. The field name prefixes the actual logged information and shows the type of information that follows on that line.Lines within a group can be in any order.

The configured logging level determines what fields are logged for a specified operation.

```
operation-indicator
field-name logged-data
            .             .
            .             .
            .             .
field-name logged-data
```

For example, the following lines can be logged by the Service Router (at logging level 5) when it routes a message.

```
routing                                          operation indicator
time 696681404 Wed Jan 29 10:36:44 2002 +0       field name, field value
type 0 message
ua-message-id P000009b00265b09
mta-message-id P000009b00265b09
message-size 209
```

O/R Address Format

O/R Address information is logged in the Scalix compact format using forward slashes (/) and commas (,). For example:

```
Chris Wolf / uk, lab/CN=Chris Wolf
```

Date and Time Format

Dates and times are always logged in the same format. Local time is used. Also recorded is the Universal Time Coordinate (UTC) offset and the number of seconds since the epoch (00:00:00 UTC January 1 1970).

```
seconds day mth dy hh:mm:ss year tz
```

Value	Description
seconds	The number of seconds since the epoch (00:00:00 UTC Jan 1 1970)
day	The day of the week (3 letters)
mth	The month (3 letters)
dy	The day of the month (1 or 2 digits)
hh	The hour (2 digits)
mm	Minutes (2 digits)
ss	Seconds (2 digits)
year	The year (4 digits)
tz	The time-zone (as +/- minutes offset from UTC)

Character Mapping in the Audit Log

When data is written to the audit log file, some character mapping is performed. This character mapping only applies to the data being logged; it does not apply to the terminating new-line of a record.

ASCII Character	Mapped As
Backspace (decimal 08)	\b
Tab (decimal 09)	\t
Newline (decimal 10)	\n
Form Feed (decimal 12)	\f
Carriage Return (decimal 13)	\r
Escape (decimal 27)	\e
Single quote (')	\'
Double quote (")	\"
Backward slash (\ )	\\

Other ASCII characters 0 through 31, and 127 are written as *nnn*, where *nnn* is the octal representation of the character. All other characters, including the space character, are written as normal.

Audit Information Logged

The following information describes the fields that can be logged to the Audit Log. Information in the Audit Log is grouped by the operation that generated the information. The groups are identified by *operation indicators*.

Each group of lines identified by an operation indicator can contain one or more lines identified by a field name followed by the logged information.

Operation Indicator	Operation
bulletin	The Bulletin Board (Public Folder) Server has received a message for a Public Folder. See “Bulletin Board Server Audit Log Information” on page 279.
delivery	The Local Delivery Service has placed a message in the Message Store. See “Local Delivery Service Audit Log Information” on page 284.
dirsync-in	The Directory Synchronization Server has received a message. See “Directory Synchronization Server Audit Log Information” on page 279.
dirsync-out	The Directory Synchronization Server has sent a message. See “Directory Synchronization Server Audit Log Information” on page 279.
omscan	The `omscan` command has been executed. See “omscan Command Audit Log Information” on page 287.
request	The Request Server has processed a request. See “Request Server Audit Log Information” on page 289.
routing	The Service Router has routed a message, acknowledgment, reply, non-delivery report, or probe. See “Service Router Audit Log Information” on page 290.
search	The Search Server has received, actioned, or deleted a request to search the Message Store. See “Search Server Audit Log Information” on page 290.
SMTP-relay	The SMTP Relay has received a message. See “SMTP Relay” on page 28.
subsystem-start	A sub-system (service) of Scalix has been started or enabled by the `omon` command. See “Scalix Service Audit Log Information” on page 288.
subsystem-stop	A sub-system (service) of Scalix has been stopped or disabled by the `omoff` command or the `omadmin` command. See “Scalix Service Audit Log Information” on page 288.
unix-in	The incoming Internet Mail Gateway has received a message. See “Internet Mail Gateway Audit Log Information” on page 282.
unix-out	The outgoing Internet Mail Gateway has sent a message. See “Internet Mail Gateway Audit Log Information” on page 282.

The ~/sys/audit.cfg file lists the fields that can be logged under a specific operation indicator. The following sections describes those fields grouped by their operation indicator.

Bulletin Board Server Audit Log Information

The following table lists the fields logged to the Audit Log when the Bulletin Board (Public Folder) Server receives a message for a Public Folder. This information in the Audit Log is grouped under the bulletin operation indicator.

Field	Description
time time-value	The *time-value* is the time at which this line was written to the file.
ua-message-id ua-message-id	The *ua-message-id* is a unique string that identifies the body of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
mta-message-id mta-message-id	The *mta-message-id* is a unique string that identifies the envelope of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
ua-ack-id ua-acknowledgment-id	The *ua-acknowledgment-id* is a unique string that identifies the body of the message that this message is acknowledging. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
mta-ack-id mta-acknowledgment-id	The *mta-acknowledgment-id* is a unique string that identifies the envelope of the message that this message is acknowledging. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
subject subject-value	The *subject-value* is the first 60 characters of the message subject. If the message is marked as "private", then this field is not logged.
originator user-name	The *user-name* is the O/R Address of the message originator.
bulletin-name bulletin-board-name	The *bulletin-board-name* is the name of the Public Folder specified in the DDA of the Bulletin Board (Public Folder) Server O/R Address.
bulletin-type bulletin-board-message-type	The *bulletin-board-message-type* is the type of the Public Folder message, which is specified as the given name of the Bulletin Board (Public Folder) Server O/R Address. The only type is "user".

Directory Synchronization Server Audit Log Information

This section describes the fields logged to the Audit Log when the Directory Synchronization Server receives or sends a message.

Incoming Directory Synchronization Server

The following table lists the information in the Audit Log grouped under the dirsync-in operation indicator.

Field	Description
time time-value	The *time-value* is the time when this line was written to the file.
originator user-name	The *user-name* is the O/R Address of the message originator. If the message is a Non-Delivery Report, this field is not logged.
subject subject-value	The *subject-value* is the first 60 characters of the message subject.
operation operation-value	The *operation-value* corresponds to a Directory Synchronization message type and is one of the following: REQUEST SSRESPONSE REPLY INDICATION FAULT
command command	The *command* corresponds to a Directory Synchronization message subtype and is one of the following: UPDATES ALL RELOAD
sending-ref reference-number	The *reference-number* is an integer identifying the Request message and is returned unchanged in the Response, Reply, or Fault message.
export-directory Directory-name	The *Directory-name* is the name of the export Directory.
import-directory Directory-name	The *Directory-name* is the name of the import Directory.
flags flag	The *flag* contains the arguments associated with the command and is one of the following: REPEAT END NONE
reload-posn position	The *position* is the temporary export reload file position.
max-reply-size size	For messages from other Scalix systems, the *size* is the maximum allowed message content size in kilobytes of any one Reply message.
syn-timestamp time-value	The *time-value* is the time entries were last imported for the Directory Synchronization agreement.
request-id id	The *id* is a string of up to 16 digits. The first 12 digits are the time the DS Server was started. The last 4 digits represent the sequence number of the message. The *id* is returned unchanged in the Reply or Fault message.

Outgoing Directory Synchronization Server

The following table lists the information in the Audit Log grouped under the `dirsync-out` operation indicator.

Field	Description
time time-value	The *time-value* is the time when this line was written to the file.
recipient user-name	The *user-name* is the O/R Address of the message recipient. If the message is a Non-Delivery Report, this field is not logged.
subject subject-value	The *subject-value* is the first 60 characters of the message subject.
operation operation-value	The *operation-value* corresponds to a Directory Synchronization message type and is one of the following: REQUEST SSREQUEST REPLY INDICATION FAULT ERROR
reload-posn position	The *position* is the temporary export reload file position.
command command	The *command* corresponds to a Directory Synchronization message subtype and is one of the following: SSUPLOADUPDATES SSUPLOADALLUPDATES ALLSSDOWNLOADUPDATES SSDOWNLOADALL RELOAD
flags flag	The *flag* contains the arguments associated with the command and is one of the following: `REPEAT END NONE`
export-directory Directory-name	The *Directory-name* is the name of the export Directory.
import-directory Directory-name	The *Directory-name* is the name of the import Directory.
syn-timestamp time-value	The *time-value* is the time entries were last imported for the Directory Synchronization agreement.
max-reply-size size	For messages from other Scalix systems, the *size* is the maximum allowed message content size in kilobytes of any one Reply message.
sending-ref reference-number	The *reference-number* is an integer identifying the Request message and is returned unchanged in the Response, Reply, or Fault message.
request-id id	The *id* is a string of up to 16 digits. The first 12 digits are the time the DS Server was started. The last 4 digits represent the sequence number of the message. The *id* is returned unchanged in the Reply or Fault message.

Internet Mail Gateway Audit Log Information

This section describes the fields logged to the Audit Log when the Internet Mail Gateway receives or sends a message.

Incoming Internet Mail Gateway

The following table lists the information in the Audit Log grouped under the `unix-in` operation indicator.

Field	Description
time time-value	The *time-value* is the time at which this line was written to the file.
ua-message-id ua-message-id	The *ua-message-id* is a unique string that identifies a message. It is generated from the internet-message-id by the `unix.in` process.
unix-message-id internet-message-id	The *internet-message-id* is a unique string generated by the internet mail system that identifies the message.
subject subject-value	The *subject-value* is the first 60 characters of the message subject. (If the subject is sensitive, this field is not logged.)
originator user-name	The *user-name* is the ARPA Address of the message originator. If the message is a Delivery Report or Non-Delivery Report, this field is not logged.
recipient-from user-name	The *user-name* is the O/R Address of a designated sender of the message (authorizing user). There can be zero, one, or more instances of this field.
recipient-to user-name	The *user-name* is the O/R Address of a primary recipient of the message. There can be zero, one, or more instances of this field.
recipient-cc user-name	The *user-name* is the O/R Address of a copy-recipient of the message (secondary recipient). There can be zero, one, or more instances of this field.
recipient-bcc user-name	The *user-name* is the O/R Address of a secondary-recipient of the message whose name is not disclosed to other recipients of the message. There can be zero, one, or more instances of this field.
orig-recipient-addr postal-address	The *postal-address* is the postal address of the originator. This field is logged if a message requires physical delivery, or if any recipient address in the message contains postal address attributes.

Outgoing Internet Mail Gateway

The following table lists the information in the Audit Log grouped under the `unix-out` operation indicator.

Field	Description
`time` time-value	The *time-value* is the time at which this line was written to the file.

Field	Description
`type` message-type message-type-name	The *message-type* and *message-type-name* can be one of the following pairs: 0 message 1 reply 2 return-of-contents 100 non-delivery-report 110 acknowledgment 120 probe 200 schedule-request
ua-message-id ua-message-id	The *ua-message-id* is a unique string that identifies the body of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
mta-message-id mta-message-id	The *mta-message-id* is a unique string that identifies the envelope of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
ua-ack-id ua-acknowledgment-id	The *ua-acknowledgment-id* is a unique string that identifies the message that this Delivery Report is acknowledging. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
hop-count hop-count-value	The *hop-count-value* is the number of store-and-forward hops the message has made. Some systems do not preserve the hop count, therefore the *hop-count-value* can be the number of hops since the message passed through such a system.
unix-message-id internet-message-id	The *internet-message-id* is a unique string, generated by the internet mail system, that identifies the message.
unix-ack-id internet-acknowledgment-id	The *internet-acknowledgment-id* is a unique string, generated by the internet mail system, that identifies the message that this Delivery Report is acknowledging.
`subject` subject-value	The *subject-value* is the first 60 characters of the message subject. If the message is marked as "private", then this field is not logged.
originator user-name	The *user-name* is the O/R Address of the message originator. If the message is a Delivery Report or Non-Delivery Report, this field is not logged.
`recipient-from` user-name	The *user-name* is the O/R Address of a designated sender of the message (authorizing user). There can be zero, one, or more instances of this field.
recipient-to user-name	The *user-name* is the O/R Address of a primary recipient of the message. There can be zero, one, or more instances of this field.
`recipient-cc` user-name	The *user-name* is the O/R Address of a copy-recipient of the message (secondary recipient). There can be zero, one, or more instances of this field.
`recipient-bcc` user-name	The *user-name* is the O/R Address of a secondary-recipient of the message whose name is not disclosed to other recipients of the message. There can be zero, one, or more instances of this field.

Field	Description
non-delivery-reason non-delivery-reason	The *non-delivery-reason* can be one of the following: 1 Routing loop 2 Operation loop 3 Message corrupt 4 Message part conversion error 5 Transport failed 6 MTA congestion 7 Too many recipients 100 Name not found at destination 101 No route for address 102 Message too big 103 Name not unique at destination 104 Address resolution failure 105 Name resolution failure 106 User Agent unavailable for delivery 107 Message expiry time reached 108 Access to service denied 109 Access to Public Distribution List denied 200 Message unable to be mapped at gateway 201 Name could not be mapped at gateway 202 Address could not be mapped at gateway 203 No conversion available for message body part 204 Mandatory message element could not be mapped 205 Conversion prohibited for this message but conversion required at gateway
orig-recipient-addr postal-address	The *postal-address* is the postal address of the originator. This field is logged if a message requires physical delivery, or if any recipient address in the message contains postal address attributes.

Local Delivery Service Audit Log Information

The following table lists the fields logged to the Audit Log when the Local Delivery Service places a message in the Message Store. This information in the Audit Log is grouped under the `delivery` operation indicator.

Field	Description
time time-value	The *time-value* is the time at which this line was written to the file.
type message-type message-type-name	The *message-type* and *message-type-name* can be one of the following pairs: 0 message 1 reply 2 return-of-contents 100 non-delivery-report 101 non-receipt-notification 110 acknowledgment 120 probe 200 Schedule-request

Field	Description
ua-message-id ua-message-id	The *ua-message-id* is a unique string that identifies the body of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
ua-ack-id ua-acknowledgment-id	The *ua-acknowledgment-id* is a unique string that identifies the body of the message that this message is acknowledging. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string*if the message has not left Scalix.
mta-message-id mta-message-id	The *mta-message-id* is a unique string that identifies the envelope of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
mta-ack-id mta-acknowledgment-id	The *mta-acknowledgment-id* is a unique string that identifies the envelope of the message that this message is acknowledging. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string*if the message has not left Scalix.
new-ua-message-id new-ua-message-id	The *new-ua-message-id* is a unique string that identifies the body of a message generated by the Local Delivery Service as a result of an automatic action performed on an incoming message.
subject subject-value	The *subject-value* is the first 60 characters of the message subject. If the message is marked as "private", then this field is not logged.
sensitivity sensitivity-level sensitivity-name	The *sensitivity-level* and *sensitivity-name* can be one of the following pairs: 0 normal 1 personal 2 private 3 company-confidential
priority priority-level priority-name	The *priority-level* and *priority-name* can be one of the following pairs: 0 normal 1 non-urgent 2 urgent
importance importance-level importance-name	The *importance-level* and *importance-name* can be one of the following pairs: 0 normal 1 low 2 high
create-time time-value	The *time-value* is the time at which this line was written to the file.
originator user-name	The *user-name* is the O/R Address of the message originator. If the message is a Delivery Report or Non-Delivery Report, this field is not logged.
recipient-from user-name	The *user-name* is the O/R Address of a designated sender of the message (authorizing user). There can be zero, one, or more instances of this field.
recipient-to user-name	The *user-name* is the O/R Address of a primary recipient of the message. There can be zero, one, or more instances of this field.

Field	Description
recipient-cc user-name	The *user-name* is the O/R Address of a copy-recipient of the message (secondary recipient). There can be zero, one, or more instances of this field.
recipient-bcc user-name	The *user-name* is the O/R Address of a secondary-recipient of the message whose name is not disclosed to other recipients of the message. There can be zero, one, or more instances of this field.
redirected-to user-name	The *user-name* is the O/R Address to which the message has been redirected.
auto-notify auto-file-value	The *auto-notify-value* is TRUE if the Local Delivery Service automatically generated a Non-Delivery Report for a message that could not be delivered.
auto-reply auto-file-value	The *auto-reply-value* is TRUE if the Local Delivery Service automatically generated a reply to the incoming message.
auto-file auto-file-value	The *auto-file-value* is TRUE if the Local Delivery Service automatically filed the incoming message in the recipient's Filing Cabinet (a container in the Message Store).
auto-print auto-print-value	The *auto-print-value* is TRUE if the Local Delivery Service automatically printed the incoming message.
auto-forward auto-forward-value	The *auto-forward-value* is TRUE if the Local Delivery Service automatically forwarded the incoming message.
delivered-count number-of-recipients	The *number-of-recipients* is the total number of local recipients the message was successfully delivered to.

Field	Description
non-delivery-reason non-delivery-reason	The *non-delivery-reason* can be one of the following: 1 Routing loop 2 Operation loop 3 Message corrupt 4 Message part conversion error 5 Transport failed 6 MTA congestion 7 Too many recipients 100 Name not found at destination 101 No route for address 102 Message too big 103 Name not unique at destination 104 Address resolution failure 105 Name resolution failure 106 User Agent unavailable for delivery 107 Message expiry time reached 108 Access to service denied 109 Access to Public Distribution List denied 200 Message unable to be mapped at gateway 201 Name could not be mapped at gateway 202 Address could not be mapped at gateway 203 No conversion available for message body part 204 Mandatory message element could not be mapped 205 Conversion prohibited for this message but required at gateway 206 Message could not be downgraded at the gateway 207 Security attributes set 208 Badly configured gateway 209 1984 X.400 addresses not supported by the gateway 270 Language is not valid for Print Server 271 Printer not configured for use by the Print Server 272 Printer error. Message could not be printed 273 Access denied to printer

omscan Command Audit Log Information

The following table lists the fields logged to the Audit Log when the `omscan` command is executed. This information in the Audit Log is grouped under the `omscan` operation indicator.

Field	Description
time time-value	The *time-value* is the time at which this line was written to the file.
user user-name	The *user-name* is the O/R Address of the user whose trays (containers) are being reported on.

Field	Description
tray tray-number tray-name number-of-messages tray-size	The *tray-number* and *tray-name* can be one of the following pairs: 1 in-tray 2 out-tray 3 pending-tray 4 filing-cabinet 5 list-area The *number-of-messages* is the total number of messages held in the tray (container). The *tray-size* is the total size in bytes of the messages held in the tray.
user-messages number-of-messages	The *number-of-messages* is the total number of messages held by the user in all their trays (containers).
user-size user-size-value	The *user-size-value* is the total size in bytes of all the messages held by the user in all their trays (containers).
bb-area-messages number-of-messages	The *number-of-messages* is the total number of messages held in the Public Folder area.
bb-area-size bb-area-size-value	The *bb-area-size-value* is the total size in bytes of all the messages held in the Public Folder area.
duration duration-value	The *duration-value* is the elapsed time, in seconds, of the `omscan` run.

Scalix Service Audit Log Information

This section shows the fields logged to the Audit Log when you start/enable or stop/disable a Scalix subsystem (service).

Subsystem Start

The following table lists the information in the Audit Log grouped under the subsystem-start operation indicator.

Field	Description
time time-value	The *time-value* is the time at which this line was written to the file.
subsystem subsystem-id subsystem-name	The *subsystem-id* is the unique identifier allocated to a service. It is usually a number but can also be a string. The remainder of the line is the *subsystem-name* - the local language name of the service.
user real-unix-name	The *real-unix-name* is the Unix user name of the person starting the subsystem.

Subsystem Stop

The following table lists the information in the Audit Log grouped under the subsystem-stop operation indicator.

Field	Description
time time-value	The *time-value* is the time at which this line was written to the file.
subsystem subsystem-id subsystem-name	The *subsystem-id* is the unique identifier allocated to a service. It is usually a number but can also be a string. The remainder of the line is the *subsystem-name*-the local language name of the service.
duration duration-value	The *duration-value* is the time (in seconds) that the subsystem was enabled.
user real-linux-name	The *real-linux-name* is the Linux user name of the person stopping the subsystem.

Request Server Audit Log Information

The following table lists the fields logged to the Audit Log when the Request Server processes a request. This information in the Audit Log is grouped under the request operation indicator.

time time-value	The *time-value* is the time at which this line was written to the file.
ua-message id ua message id	The *ua-message-id* is a unique string that identifies the body of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
mta-message id mta-message-id	The *mta-message-id* is a unique string that identifies the envelope of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
ua-ack id ua-acknowledgment-id	The *ua-acknowledgment-id* is a unique string that identifies the body of the message that this message is acknowledging. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
originator user-name	The *user-name* is the O/R Address of the message originator.
mta-ack-id mta-acknowledgment-id	The *mta-acknowledgment-id* is a unique string that identifies the envelope of the message that this message is acknowledging. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
originator-admin-cap originator-admin-cap	The *originator-admin-cap* is TRUE if the originator of the message has Scalix Administration capability, or FALSE if the originator does not have Scalix Administration capability.

subject subject-value	The *subject-value* is the first 60 characters of the message subject. If the message is marked as "private", then this field is not logged.
request-name request-name	The *request-name* is the name of the request, which is specified as the given name of the Request Server O/R Address.

Search Server Audit Log Information

The following table lists the fields logged to the Audit Log when the Search Server receives, actions, or deletes a request to search the Message Store. This information in the Audit Log is grouped under the search operation indicator.

Field	Description
time time-value	The *time-value* is the time at which this line was written to the file.
received-request request-reference	The *request-reference* is the direct reference of the search request transaction file containing the criteria for the items to be searched for in the Message Store.
deleted-request request-reference	The *request-reference* is the direct reference of the search request transaction file deleted by the UAL client.
started-request request-reference	The *request-reference* is the direct reference of the search request transaction file when the search is started.
completed-request request-reference	The *request-reference* is the direct reference of the search request transaction file when the search is complete.
hits number-of-hits	The *number-of-hits* is the number of items that match the search criteria specified in the search request transaction file.

Service Router Audit Log Information

The following table lists the fields logged to the Audit Log when the Service Router routes a message, acknowledgment, reply, non-delivery report, or probe. This information in the Audit Log is grouped under the routing operation indicator.

Field	Description
time time-value	The *time-value* is the time at which this line was written to the file.

Field	Description
type message-type message-type-name	The message-type and message-type-name can be one of the following pairs: 0 message 1 reply 2 return-of-contents 100 non-delivery-report 101 non-receipt-notification 110 acknowledgment 120 probe 200 schedule-request
ua-message-id ua-message-id	The *ua-message-id* is a unique string that identifies the body of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
mta-message-id mta-message-id	The *mta-message-id* is a unique string that identifies the envelope of a message. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
ua-ack-id ua-acknowledgment-id	The *ua-acknowledgment-id* is a unique string that identifies the body of the message that this message is acknowledging. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string* if the message has not left Scalix.
mta-ack-id mta-acknowledgment-id	The *mta-acknowledgment-id* is a unique string that identifies the envelope of the message that this message is acknowledging. It takes the form *id-string-O/R-Address* if the message has passed through an X.400 Interface, or *id-string*if the message has not left Scalix.
subject subject-value	The *subject-value* is the first 60 characters of the message subject. If the message is marked as "sensitive", then this field is not logged.
sensitivity sensitivity -level sensitivity name	The *sensitivity-level* and *sensitivity-name* can be one of the following pairs: 0 normal 1 personal 2 private 3 company-confidential
priority priority-level priority-name	The *priority-level* and *priority-name* can be one of the following pairs: 0 normal 1 non-urgent 2 urgent
Importance importance-level importance-name	The *importance-level*and *importance-name* can be one of the following pairs: 0 normal 1 low 2 high
created-locally created-locally	The *created-locally* is 1 if the message was created on the local Scalix Server; 0 if the message was created on a remote Scalix Server.
originator user-name	The *user-name* is the O/R Address of the message originator. If the message is a Delivery Report or Non-Delivery Report, this field is not logged.

Field	Description
designate-originator user-name	The *user-name* is the O/R Address of the message originator, who is a designate.
part-type part-type-value type-string	The *part-type-value* is a number representing the Scalix item type. The *type-string* is the name of the associated item type taken from the file `~/nls/language/filetype`. Scalix uses the most appropriate filetype file for the language used by the Service Router when it is started.
part-size part-size-value	The *part-size-value* is the size in bytes of that part of the message.
message-size message-size-value	The *message-size-value* is the total size in bytes of the message.
part-count number-of-parts	The *number-of-parts* is the total number of body parts in the message.
recipient-from user-name	The *user-name* is the O/R Address of a designated sender of the message (authorizing user). There can be zero, one, or more instances of this field.
recipient-to user-name	The *user-name* is the O/R Address of a primary recipient of the message. There can be zero, one, or more instances of this field.
recipient-cc user-name	The *user-name* is the O/R Address of a copy-recipient of the message (secondary recipient). There can be zero, one, or more instances of this field.
recipient-bcc user-name	The *user-name* is the O/R Address of a secondary-recipient of the message whose name is not disclosed to other recipients of the message. There can be zero, one, or more instances of this field.
queue queue-name	The *queue-name* is the name of the queue that the Service Router attaches the message to. There can be several instances of *queue-name* for different recipients on different systems. The *queue-name* can be followed by queue-specific service information.
hop-count hop-count-value	The *hop-count-value* is the number of store-and-forward hops the message has made. Some systems do not preserve the hop count, therefore the *hop-count-value* can be the number of hops since the message passed through such a system.

Field	Description
non-delivery-reason non-delivery-reason	The *non-delivery-reason* can be one of the following: 1 Routing loop 2 Operation loop 3 Message corrupt 4 Message part conversion error 5 Transport failed 6 MTA congestion 7 Too many recipients 100 Name not found at destination 101 No route for address 102 Message too big 103 Name not unique at destination 104 Address resolution failure 105 Name resolution failure 106 User Agent unavailable for delivery 107 Message expiry time reached 108 Access to service denied 109 Access to Public Distribution List denied 200 Message unable to be mapped at gateway 201 Name could not be mapped at gateway 202 Address could not be mapped at gateway 203 No conversion available for message body part 204 Mandatory message element could not be mapped 205 Conversion prohibited for this message but required at gateway 206 Message could not be downgraded at the gateway 207 Security attributes set 208 Badly configured gateway 209 1984 X.400 addresses not supported by the gateway 270 Language is not valid for Print Server 271 Printer not configured for use by the Print Server 272 Printer error. Message could not be printed 273 Access denied to printer
max-nest-depth depth	The *depth* is a number giving the maximum depth of nesting in a message. A flat message will have a *depth* of 0.
ack-req acknowledgment-number acknowledgment-name	The *acknowledgment-number* and *acknowledgment-name* is the acknowledgment that has been requested on the message to a particular recipient. It can be one of the following pairs: 0 none 1 transmit 2 receive 3 non-delivery 4 delivery 5 auto-forward 6 delete 7 read 8 auto-answer 9 reply
all-recips-non-deliv non-delivery-reason	The *non-delivery-reason* is as above. In this case, all recipients in the active recipient list do not get the message, usually due to a looping problem.

Field	Description
delivered-count number-of-recipients	The *number-of-recipients* is the number of recipients the message was successfully routed to.

SMTP Relay Audit Log Information

The following table lists the fields logged to the Audit Log when the SMTP Relay receives a message. This information in the Audit Log is grouped under the SMTP-relay operation indicator.

Field	Description
time time-value	The time this record was logged.
mta-message-id message-id	The message ID of the message being processed.
originator-domain originator-domain	The domain of the originating SMTP connection (`HELO` SMTP command).
originator originator	The Internet address of the originator (`MAIL FROM` SMTP command).
authentication	Authentication information.
recipient-to internet-address	The Internet address of a recipient (`RCPT TO` SMTP command).
recipient-target [U] [S] [X]	The system to which the above recipient is relayed: [U]: `unix.in` [S]: `sendmail` [X]: `xport.in`
summary-target [U] [S] [X]	Summary of all the systems to which the message is relayed. [U]: `unix.in` [S]: `sendmail` [X]: `xport.in`
hop-count hop-count	The number of hops the message has made so far.
message-size message-size	The size of the message in bytes.

Audit Log Commands

The following table lists and describes commands associated with the Audit Log.

Command	Description
omconfaud	Configures Audit Log logging levels. For example, enter: `omconfaud router 11` to configure an Audit level of `11` for the Service Router. Enter: omconfaud -a imap 9 to configure an Audit level of 9 for the IMAP Server daemon.
omshowaud	Shows Audit Log logging levels. The logging level displays for the following services: Archiver Service Router Local Delivery Internet Mail Gateway Local Client Interface Remote Client Interface Administration Request Server Directory Synchronization Bulletin Board Server Background Search Service POP3 interface Omscan Server Enter `omshowaud -a`. The logging level displays for the following daemons: Server Daemon PC Monitor Directory Relay Server Notification Server Shared memory daemon Notification Monitor Session Monitor Container Access Monitor Item Structure Server Database Monitor Licence Monitor Daemon LDAP Daemon Queue Manager Item Delete Daemon IMAP Server Daemon SMTP Relay Mime Browser Controller

Chapter 23

Error Manager

The Error Manager receives copies of Non-delivery Reports generated by the Scalix Servers that the Error Manager is responsible for. The Error Manager also receives messages addressed to the Error Manager Server (surname +ERRMGR). This chapter includes the following information:

Error Manager Overview

Using the information contained in the Non-Delivery Reports, the user who is designated as the Error Manager can identify and resolve any network and routing problems that occur.

The Error Manager can be any Scalix user. The user can reside on the local machine or on a remote machine. A single user can be the Error Manager for a number of Scalix Servers. If the Error Manager is not local to an Scalix Server they are responsible for, there is a risk that routing problems may prevent the delivery of Non-Delivery Reports.

Usually, the user designated as the Error Manager is the Scalix Administrator for the local system. However, to avoid mixing Non-Delivery Reports with personal messages, a fictitious user can be added to the system as the Error Manager. The Administrator can then check each day the fictitious user's In Tray for Non-Delivery Reports and other messages sent to the Error Manager.

The Error Manager O/R Address is configured using the omconfenu command and displayed using the omshowenu command.

An Error Manager must be designated for every Scalix Server in a network. If an Scalix Server does not have an Error Manager, the server logs non-delivery information in the Event Log as a "Warning"; no Non-Delivery Reports will be generated for the Error Manager.

Use the General Configuration File (~/sys/general.cfg) option NDN_EM_SERIOUS_ONLY to limit the number of Non-Delivery Reports that are sent to the Error Manager. This option stops Scalix returning Non-Delivery Reports for simple mis-addressing problems to the Error Manager. Non-Delivery Reports of this kind are returned to the originator only.

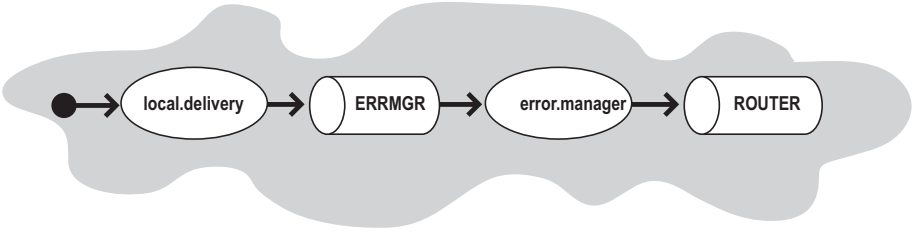
You can also use the option NDN_NO_ALTERNATES to stop alternate names being placed in a Non-Delivery Report.

For more information, see “Configuration Options” on page 301.

Error Manager Server

The Error Manager Server routes messages to the designated Error Manager for a local system. The Error Manager Server provides a way of contacting a remote System Administrator whose address you do not know.

A user can send a message addressed to any Error Manager Server in the system. The Error Manager Server then sends the message on to the designated Error Manager for that system.



- Messages addressed to the Error Manager Server are routed in the normal way by the Service Router.

  If the message is for a local address, it is passed to the Local Delivery Service by the Service Router.
- The Local Delivery Service attaches messages addressed to the user +ERRMGR to the Error Manager Server queue ERRMGR.
- The Error Manager Server replaces the active recipient list with the O/R Address of the Error Manager.

  (The active distribution list remains unchanged so that the message, when displayed, will still appear to be addressed to the user +ERRMGR rather than the actual Error Manager.)
- The message is then routed through the Service Router for transmission and delivery in the normal way.

The Error Manager Server is enabled by the Local Delivery Service when Local Delivery is started.

The following summarizes key Error Manager Server information.

- Service name: The Error Manager Server does not have a service name as it is automatically enabled by the Local Delivery Service.
- Process name: error.manager
- Queue name: ERRMGR
- Surname: +ERRMGR

Configuration and Addressing

The following describes configuration and addressing issues.

If the Error Manager User is not Configured

When the Error Manager Server is enabled by the Local Delivery Service, it reads the address of the Error Manager.

If an Error Manager is not configured, the server will log this in the Event Log as a "Warning". No messages will be processed by the Error Manager Server, although messages will still be attached to its queue ERRMGR.

Addressing the Error Manager Server

The surname of the Error Manager Server is +ERRMGR. This entity (user) appears to Scalix to exist at all mailnodes in the system.

For example, after sending a test message to an address and receiving back a Non-Delivery Report stating that there is a routing problem on a system with a primary mailnode of boston,factory,mis, you can contact the Error Manager on that system by sending a message addressed to:

```
+ERRMGR/boston,factory,mis
```

For additional information on the primary mailnode, see the section “Primary Mailnode” on page 6.

Error Manager Commands

The following table describes commands associated with the Scalix Error Manager.

Command	Description
omconfenu	Configure an Error Manager
omshowenu	Show the address of the Error Manager

Chapter 24

Configuration Options

This chapter describes configuration options you modify in configuration files to customize the Scalix Server. This chapter includes the following information:

Configuration Files

The general configuration files contain options that affect the behavior of the Scalix system. You can modify system-wide, client-specific, and user-specific configuration options.

Scalix includes a number of hard-coded default options. You can change these options by placing TAG=value pairs in one or more of the following configuration files listed in the following table in the ~/scalix/sys/ directory.

File Name	Description
general.cfg	Configuration options affecting all Scalix system users.
client.cfg/*hostname*	A subset of configuration options affecting the individual Scalix client for which they are specified.
user.cfg/scalix-*ID*	A subset of configuration options affecting the individual Scalix user for whom they are specified.
route.cfg/*routename*	A subset of configuration options affecting the route for which they are specified.
lang.cfg/*language*	A subset of configuration options affecting the language for which they are specified.

Note that any values that contain underscores (_) or spaces should be specified within double quotes, for example:

```
EXAMPLE_OPTION="one_two three"
```

Some options can be set in more than one of the above files. In this case, note that user-specific options override client-specific options, and client-specific options override general options.

System-wide Configuration Options

Option	Description
ARCHIVE=TRUE NOTE: Because all of these settings include modifications to the general.cfg file, you must restart the service router and archiver for the changes to take effect.	Enables the archiving of all messages that traverse the Scalix Server. Messages are archived to the `~scalix/archive` directory. This parameter includes the following parameters: FALSE Disables the Archiver. • arch:/*path*/ This archives messages to the directory you specify. Within this directory, the Archiver automatically creates subdirectories based on the date that messages traverse the Scalix Server. For example, if a message arrives at 11:55 on June 10th, 2004, the message is archived to: /*path*/2004-06-10/14:55+0000.12345.1 `where +0000` is the local time zone offset from GMT. `and 12345` is the PID of the Archiver. `.1` indicates the number of messages that arrived during that second. NOTE: The Archiver process operates as the user "`scalix`". If you want the Archiver to archive messages to the `/home` directory, you must configure the permissions for the `/home` directory to allow the `scalix` user write access (to this directory). • inet:host.example.com inet:host.example.com:2000 This allows connection to the host on port 25 or on a port you specify (`host.example.com:2000`). This creates an SMTP session and enables you to use third-party archiving systems. • ARCHIVE=bcc:archive@example.com This forwards to a designated "bcc" mailbox created solely for archiving purposes a blind (bcc) copy of every message that is sent. You must create this mailbox before adding this parameter. We recommend it be on a separate box because archive files use significant memory space. • file:/*path*/*archive_file_name* This writes all messages to a single file. You cannot use this option with auxiliary processes. • fork:/bin/archive_script.sh The archiver forks the script and communicates with SMTP using stdin and stdout to the script.
ARCH_TNEF_ENCODE=TRUE	Sets TNEF as the message format for archived messages. By default, the Archiver converts messages to MIME format and consequently looses some MAPI information (if applicable).

Audit Log options

Option	Description
AUD_88_NAMES=TRUE	Sets audit logging on for the X.400 1988 address attributes. By default, only the 1984 attributes are logged.
AUD_LOG_UX_NAME=FALSE	By default, users are identified in the Audit Log by their Scalix IDs. Set this option to TRUE if you want users to be identified by their Linux user names.

Auto Actions options

Option	Description
AA_DEFAULT_LOGGING_ON=TRUE	Sets the default setting for logging of automatic actions to on for all users.
AA_GLOBAL_LOGGING_OFF=TRUE	Stops all logging of automatic actions even if configured in a client. AA_GLOBAL_LOGGING_OFF overrides AA_DEFAULT_LOGGING_ON if it is set.
AA_MAXCFG_LOG_SIZE=*size_in_bytes*	Sets, for all users, the maximum size of their automatic action log file. The maximum size is 65536 bytes.
FLT_ESC_NO_CONV=TRUE	If set to TRUE, a serious error is reported when the character set for a filter and that for a string being filtered are not of the same kind while filtering for automatic actions. If set to FALSE, this is reported as a failed match.

Client Directory Access Server Options

Option	Description
CDA_CHECKTIME=*minutes*	Set the time interval, in minutes, at which the CDA Server checks its configuration for Directories that need processing. The default is 5 minutes.
CDA_USE_CHANGE_LOG=TRUE	Set this option to optimize the rebuilding of Directory access tables by the CDA Server. By default, the CDA Server will rebuild the access tables periodically. If CDA_USE_CHANGE_LOG=TRUE is set, the CDA Server first checks the change log for the Directory, and only rebuilds the access tables if the change log shows that the Directory has been modified. If the Directory does not have a change log, the CDA Server will configure one.

Daemon Options

Option	Description
DADM_DAEMON_TIME_TO_EXIT= *seconds*	The number of seconds Scalix will wait for a daemon to exit before sending a `SIGKILL` signal to stop the daemon process. The default is 30 seconds; if daemon processes are being stopped too quickly, increase this number. Use this option with caution as the `SIGKILL` signal does not allow the daemon process to tidy up before it stops.

Directory Options

Option	Description
CU_LOG_OLD_DIR_CMDS=TRUE	If set, the old Directory commands `omadddir` and `ommoddir` are used when logging changes to the Directory update file (`om_record`), instead of the new `omaddent`,`omdelent`, and `ommodent` commands.
DA_IGNORE_INDEXES=*attribute,attribute,...*	Under normal circumstances, Scalix will fail to locate a match for any Scalix attribute which is keyed but for which the index does not exist. Use this option to specify those Scalix attributes that are keyed but for which Scalix should search sequentially, rather than attempt to use the indexes for the attributes. *attribute,attribute,...* is a comma-separated list of Scalix internal attribute names. Do not insert spaces after the commas. This option can be used for newly keyed attributes for which the indexes have not yet been built.
DIR_IA_UNIQUECHECK_OFF= *directory-list*	Use this option to specify those directories in which uniqueness checking of the Internet address attributes is turned off. The value of this option is a comma-separated list of Scalix directories or `ALL`.
DIR_IA_UNIQUECHECK_ON= *directory-list*	Use this option to specify those directories in which uniqueness checking of the Internet address attributes is turned on. The value of this option is a comma-separated list of Scalix directories or `ALL`.
VI_NON_SUPP_ATTS_UNIQUE= TRUE	By default, the combined values of the O/R Address attributes (shown by an X in the `omshowatt` command) in a Directory entry must be unique. If they are not, the entry cannot be added or modified. If `VI_NON_SUPP_ATTS_UNIQUE=TRUE` is set, this rule is relaxed and the combined values of all attributes, other than supplementary attributes, are used to determine the uniqueness of the entry rather than just those of the O/R Address attributes. (The same effect can be obtained for an individual session by setting and exporting the environment variable `VI_NON_SUPP_ATTS_UNIQUE=TRUE`.)

Option	Description
VI_SORTED_VISTA_DATABASE= FALSE	Determines whether the matching entries resulting from a directory search are returned sorted or in random order. When this option is FALSE (the default), the matching entries are returned in random order. Set this option to TRUE to cause the matching entries to be sorted by key. If you set this option to TRUE, you must rebuild all the directories (using `diropt`) for it to take effect.

Directory Relay Server Options

Option	Description
DRS_HOST_RETRY_TIMEOUT= *seconds*	If the remote directory access mechanism fails to contact a remote host, it avoids retrying the connection for the number of seconds specified by this option. If a subsequent request for the same host occurs within the specified period, an error is returned immediately. The default value is 60.
DRS_MAX_CHILDREN= *number_of_child_processes*	Specifies the maximum number of processes the Directory Relay Server can support at once. Each process has a separate bind to the Directory Information Base, and some X.500 implementations can support only a limited number of binds at once. Therefore, specify the maximum number of processes with this in mind.
DRS_RESERVED_CHILDREN= *number_of_child_processes*	Defines the minimum number of child processes that the Directory Relay Server will maintain at once. Each child process has its own bind to the Directory Information Base. The greater the minimum number of child processes, the better the performance of Directory lookups. However, each process consumes resources. You should not normally set this to a value lower than 3: one process each for Local Delivery and the Service Router, and one to handle requests from the UAL and `omsearch`. The default value is 3.

Directory Synchronization Options

Option	Description
DR_NO_MOD_STRIP_PDL=TRUE	Determines the amount of information included for an entry in the Directory change log when changes are made to PDL members within the exporting Directory using the following PDL commands: omaddpdln omdelpdln ommodpdln When set to TRUE, the full PDL entry, including PDL members, will be logged in the from entry in the `MODIFY` record. Otherwise, only the X.400 attributes in the PDL entry will be logged to save disk space and improve Directory synchronization performance.
DS_ADD_UPDATE_LOCAL_ENTRY=TRUE	Determines the behavior of a Directory synchronization search. If the `ADD` operation returns a matching entry and `DS_ADD_UPDATE_LOCAL_ENTRY` is set to `TRUE`, a `MODIFY` operation is performed. During the `MODIFY` operation, the matched entry in the importing Directory is modified with the corresponding entry in the exporting Directory. If the `MODIFY` operation is successful, a `MODIFY` record is added to the Directory change log. Using this option can modify entries in the importing Directory which are genuinely different from those in the exporting Directory.
DS_CUST_IMP_ADD_UPDATE_LOCAL_ENTRY=TRUE	Determines the result of a Directory synchronization search. During Directory synchronization, an entry in the exporting Directory will match an entry in the importing Directory if the search attributes are identical, even though other attributes of the entry may be different. In this case, if this option is set to `TRUE`, the entry in the importing directory is overwritten by the corresponding entry in the exporting directory. If the option is set to `FALSE`, the entry in the exporting Directory is treated as a duplicate of the corresponding entry in the importing directory, and so is not exported. Using this option can have the effect of modifying entries in the importing Directory which are genuinely different from those in the exporting Directory.
DS_CUST_MSGQ_TIMEOUT=*seconds*	The number of seconds the DS Server will wait on an empty message queue before checking if any of its timers have expired. Defaults to 1800 seconds. Use in conjunction with `DS_CUST_PERIOD_TIMER_MINUTES` when testing Directory synchronization.
DS_CUST_PERIOD_TIMER_MINUTES=TRUE	If set, the value of the period timer for Scalix-to-Scalix Directory synchronization is in minutes. The default is for the value of the period timer to be in hours. Use in conjunction with `DS_CUST_SEND_REQ_NOW` when testing Directory synchronization.

Option	Description
DS_CUST_SEND_REQ_NOW=TRUE	If set, Request messages are sent immediately the first timer check is performed after a restart of the DS Server if the period timer value (default 24 hours) is greater than the time from the current time to the start time. The default is to wait for the period timer to expire. Use in conjunction with DS_CUST_PERIOD_TIMER_MINUTES when testing Directory synchronization.
DS_SEND_SOURCE_LID=TRUE	When set to TRUE, a unique identifier is propagated with each entry in each transaction during Directory synchronization. The value of the identifier is that of the LOCAL-UNIQUE-ID attribute of the entry in the exporting Directory. Setting this option to FALSE prevents this identifier being sent during Directory synchronization. The default setting for this option is TRUE.

Option	Description
DR_NO_MOD_STRIP_PDL=TRUE	Determines the amount of information included for an entry in the Directory change log when changes are made to PDL members within the exporting Directory using the following PDL commands: • omaddpdln • omdelpdln • ommodpdln When set to TRUE, the full PDL entry, including PDL members, will be logged in the from entry in the MODIFY record. Otherwise, only the X.400 attributes in the PDL entry will be logged to save disk space and improve Directory synchronization performance.
DS_ADD_UPDATE_LOCAL_ENTRY=TRUE	Determines the behavior of a Directory synchronization search. If the ADD operation returns a matching entry and DS_ADD_UPDATE_LOCAL_ENTRY is set to TRUE, a MODIFY operation is performed. During the MODIFY operation, the matched entry in the importing Directory is modified with the corresponding entry in the exporting Directory. If the MODIFY operation is successful, a MODIFY record is added to the Directory change log. Using this option can modify entries in the importing Directory which are genuinely different from those in the exporting Directory.

Option	Description
DS_CUST_IMP_ADD_UPDATE_ LOCAL_ENTRY=TRUE	Determines the result of a Directory synchronization search. During Directory synchronization, an entry in the exporting Directory will match an entry in the importing Directory if the search attributes are identical, even though other attributes of the entry may be different. In this case, if this option is set to TRUE, the entry in the importing directory is overwritten by the corresponding entry in the exporting directory. If the option is set to FALSE, the entry in the exporting Directory is treated as a duplicate of the corresponding entry in the importing directory, and so is not exported. Using this option can have the effect of modifying entries in the importing Directory which are genuinely different from those in the exporting Directory.
DS_CUST_MSGQ_TIMEOUT=*seconds*	The number of seconds the DS Server will wait on an empty message queue before checking if any of its timers have expired. Defaults to 1800 seconds. Use in conjunction with DS_CUST_PERIOD_TIMER_MINUTES when testing Directory synchronization.
DS_CUST_PERIOD_TIMER_ MINUTES=TRUE	If set, the value of the period timer for Scalix to Scalix Directory synchronization is in minutes. The default is for the value of the period timer to be in hours. Use in conjunction with DS_CUST_SEND_REQ_NOW when testing Directory synchronization.
DS_CUST_SEND_REQ_NOW=TRUE	If set, Request messages are sent immediately the first timer check is performed after a restart of the DS Server if the period timer value (default 24 hours) is greater than the time from the current time to the start time. The default is to wait for the period timer to expire. Use in conjunction with DS_CUST_PERIOD_TIMER_MINUTES when testing Directory synchronization.
DS_SEND_SOURCE_LID=TRUE	When set to TRUE, a unique identifier is propagated with each entry in each transaction during Directory synchronization. The value of the identifier is that of the LOCAL-UNIQUE-ID attribute of the entry in the exporting Directory. Setting this option to FALSE prevents this identifier being sent during Directory synchronization. The default setting for this option is TRUE.

IMAP Client Options

Option	Description
IMAP_AUTOMATIC_MDN=FALSE	Determines whether the IMAP4 Server should generate Message Disposition Notification messages (MDNs) automatically. RFC2298 specifies that IMAP4 clients should generate MDNs where requested. However, some clients are unable to do this. Set this option to TRUE if you wish the IMAP4 Server to send MDNs automatically, without reference to the IMAP4 client.
IMAP_BB_FOLDER_PREFIX=#bb	Specifies the string that precedes all mailbox names for Public Folders. Use this option to enable users to distinguish easily between Public Folders and private folders. You can specify any ASCII character as a separator, although you are recommended not to choose 0-9, a-z, A-Z, +, comma, &, – (minus), tab, space, newline, %, or *. You must not use the same value for this option as for the IMAP_FOLDER_PREFIX option, or the IMAP_FOLDER_SEPARATOR or IMAP_BB_FOLDER_SEPARATOR options. You must enter the character itself, rather than the ASCII code for it.
IMAP_BB_FOLDER_SEPARATOR=/	Specifies the character used to separate public folder names in an IMAP mailbox specification. For example, with a separator of / and a Public Folder beta whose parent Public Folder is alpha, the corresponding IMAP mailbox name is alpha/beta. You can specify any ASCII character as a separator, although you are recommended not to choose 0-9, a-z, A-Z, +, comma, &, – (minus), tab, space, newline, %, or *. You must enter the character itself, rather than the ASCII code for it. You can need to use a Public Folder separator that is different from the default (/) if you wish to use this character in Public Folder names. For example, if there is an existing Public Folder called Sales/Forecasts, this Public Folder will not be seen through the IMAP Server since it could not be distinguished from a Public Folder called Forecasts within a Public Folder called Sales. There is a separate option to set the separator for private folders: see IMAP_FOLDER_SEPARATOR later in this section. See also the related options IMAP_FOLDER_PREFIX and IMAP_BB_FOLDER_PREFIX.

Option	Description
IMAP_CAPABILITIES= *capabilities-list*	Lists the capabilities that the IMAP Server advertises to the client. Each item in the list is separated by a space. The default list of capabilities is "`IMAP4 IMAP4rev1 IDLE NAMESPACE AUTH=LOGIN`". The capabilities that can be included in the list are: `IMAP4:` Support for the basic protocol as defined in RFC1730. Note that the IMAP4 commands defined in RFC1730 but absent in RFC2060 are still supported even if this capability is not advertised. `IMAP4rev1:` Support for the basic protocol as defined in RFC2060. The IMAP Server always advertises this capability. `CHILDREN:` Support for a means of indicating whether or not folders have child folders or not. This is not a standard extension. `IDLE:` Support for the `IDLE` extension as defined in RFC2177. This extension can provide a significant performance benefit for clients that can use it. `LITERAL+:` Support for non-synchronizing literals as defined in RFC2088. Use this capability with caution since it leaves the Server open to certain kinds of denial-of-service attacks. `NAMESPACE:` Support for the `NAMESPACE` command as defined in RFC2342. This command is used by certain clients to discover the namespace prefix for Public Folders so that these can be seen by the client user.
IMAP_CONNECTION LIMIT=0	Specifies the maximum number of concurrent IMAP connections that the Server can support. If left at 0 (the default value), the IMAP4 Server will continue to accept all connections until machine resources are exhausted. This could adversely affect Scalix performance and eventually prevent other users from accessing the Scalix Server.
IMAP_CONNRATE_LIMIT=0	Specifies the maximum number of IMAP connection requests that the IMAP Server will accept in any one second. If left at 0 (the default), the IMAP Server will accept connection requests at a rate that is limited only by machine resources. This could adversely affect Scalix performance and eventually prevent other users from accessing the Scalix Server. If, for example, you set this value to 3, the IMAP Server will be able to accept up to 180 connection requests per minute, and machine resources should still be sufficient to allow normal Scalix operation.
IMAP_DELETE_SUBFOLDERS= FALSE	Determines whether the IMAP4 Server permits the deletion of folders that contain subfolders. When this option is set to `FALSE` (the default), the IMAP4 Server will not allow a client to delete a folder that contains subfolders. This is in accordance with the IMAP4 protocol. However, some non-conforming clients will attempt to delete such folders. Set this option to `TRUE` if you wish to allow such attempts to succeed (and possibly enhance the usability of these clients).

Option	Description
IMAP_FOLDER_SEPARATOR=/	Specifies the character used to separate private folder names in an IMAP mailbox specification. For example, with a separator of `/` and a folder named `beta` inside another folder named `alpha`, the corresponding IMAP mailbox name is `alpha/beta`. You can specify any ASCII character as a separator, although you are recommended not to choose 0-9, a-z, A-Z, +, comma, &, – (minus), tab, space, newline, %, or `*`. You must enter the character itself, rather than the ASCII code for it. You can need to use a folder separator that is different from the default (`/`) if you wish to use this character in folder names. For example, if an existing Scalix user has a folder called `Sales/Forecasts`, this folder will not be seen through the IMAP Server since it could not be distinguished from a folder called `Forecasts` within a folder called `Sales`. There is a separate option to set the separator for Public Folders: see `IMAP_BB_FOLDER_SEPARATOR` earlier in this section. See also the related options `IMAP_FOLDER_PREFIX` and `IMAP_BB_FOLDER_PREFIX`.
IMAP_IDLE_TIMEOUT=30	Specifies the number of minutes an IMAP connection can remain idle before the connection is closed by the IMAP Server. Specify a value of 0 to disable idle timeouts. Note that the IMAP protocol (RFC2060) specifies a minimum timeout of 30 minutes. Some clients can wait exactly 30 minutes between commands and so are liable to get logged out prematurely if this option is not set, or is set to its default value. For these clients, it is sometimes useful to set the idle timeout to 31 minutes.
IMAP_IGNORE_SERVERNAME=FALSE	Determines whether the IMAP Server uses the characters following the @ character in a username as the Server name for this user. When set to `FALSE` (the default), the name part of the username (up to and including the @ character) is stripped off and the remainder is used as the Server name to which the IMAP connection is relayed. Set this option to `TRUE` to prevent the IMAP connection being relayed to another Server.
IMAP_LOGFILE=~/tmp/imap.%h	Specifies the name of the file to which IMAP events are logged, provided that logging is turned on using the `IMAP_LOGLEVEL` option. If the file you specify already exists, new events will be appended to it. Note that at certain log levels log files can contain sensitive information, such as passwords. You can use the following tokens in the log file name: `%p:` Expands to the PID of the IMAP Server process. One log file will be created for each IMAP Server process. `%h:` Expands to the name of the client host. One log file will be created for each client host that connects to the IMAP Server. `%u:` Expands to the Scalix UID. One log file will be created for each Scalix user that connects to the IMAP Server.

Option	Description
IMAP_LOGLEVEL=0	Activates logging of IMAP commands and errors. The log file is specified by the IMAP_LOGFILE option. Set a value of 0 to disable logging, 1 to log basic commands/responses only, 2 to log unexpected UAL errors, and 8 to enable raw protocol logging. Note that, at log level 8, passwords are recorded in the log files. To avoid this, you can set a lower log level in the system-wide or per-client configuration file, and then set the log level to 8 in the user-specific configuration file. This log level will only take effect after authentication, and so passwords will not be recorded. If you require several kinds of logging information, add the numbers for the log levels you require. If the option IMAP_UAL_TRACE_LEVEL is not defined, then setting IMAP_LOGLEVEL to any value other than 0 enables UAL logging for the IMAP Server.
IMAP_MAILSTORE_HOST=*hostname*	Specifies the fully qualified domain name of the Scalix host to which the IMAP Server connects. Use this option when the IMAP4 Server does not reside on the same machine as the Scalix system that contains the relevant message store.
IMAP_MDNSENT_FLAG=$MdnSent	Determines the name of the flag that is set to indicate that a Message Disposition Notification (MDN) has been sent for this message. (See also the option IMAP_AUTOMATIC_MDN.) The name of this flag has not been standardized, and therefore different IMAP4 clients can use different flag names. Set this option to the name of the flag that your IMAP clients use.
IMAP_MIN_SIZE_ESTIMATE=0	Specifies if the client will compute message sizes, or will estimate them. Some clients report the message size when they list messages in the user's In Tray. To do this, they must render the message, which can be time-consuming, and cause a decrease in performance. To prevent the client from rendering messages above a certain size, specify this size in kilobytes. For example, to prevent the IMAP client from rendering all messages above 5 kilobytes, set this value to 5. Messages less than about 5 kilobytes will be rendered and have their size reported accurately. Messages larger than about 5 kilobytes will have an estimate of their size reported. Note that some clients require the message size to be computed accurately. For these clients, you must set this option to 0 or leave it undefined.

Option	Description
IMAP_REMOTE_UAL_ENABLED= TRUE	Specifies whether an IMAP client can use a remote UAL Server. Local connections have better performance. If set to TRUE, users can specify the name of a remote machine on which is running a UAL Server. The IMAP Server will then use this remote UAL Server. Users specify the use of a remote UAL Server by connecting as *user-rname@hostname*, where *hostname* is the name of the remote machine to which they wish to connect. Set this option to FALSE if you wish to prevent users from connecting to a remote UAL Server.
IMAP_SEARCH_TIMEOUT=0	Specifies the number of seconds to wait before abandoning a search request. Specify a value of 0 to prevent search requests from timing out.
IMAP_UAL_TRACE_LEVEL=0	Activates tracing of IMAP Server information at the UAL Server. The trace files are placed in the ~/tmp directory. If this directory cannot be found, they are placed in the /tmp directory. Possible values, and the corresponding file names, are shown in the following table. *user-no* is the Scalix user number. If you require several different kinds of trace information, add the numbers for the levels you require and set the entry to the total. • 0: No tracing. The default. • 1: Raw (unformatted) command/reply tracing (file name: OM*user-no*N.trc). • 2: Symbolic command/reply tracing (file name: OM*user-no*C.trc). • 4: Message Store file name mapping. No trace file. The subject of an item listed or displayed in the client is replaced by its corresponding Message Store file name. • 8: Full tracing of command/reply and file transfer data. This can be used to rerun a session (file name OM*user-no*U.log and OM*user-no*U.f*nnnn)*. • 16: Raw (unformatted) command/reply tracing and file transfer data (*user-no*N.trc). This option is similar to the UAL_TRACE_LEVEL option. However, the UAL_TRACE_LEVEL option is user-specific, and causes information on all UAL clients to be traced. The IMAP_UAL_TRACE_LEVEL option is IMAP-specific (that is, its trace files contain information on all users of a particular machine), but it traces only IMAP information.

Internet Addressing Options

Option	Description
INET_AUTOGEN_IA_ON_MODIFY= FALSE	Determines whether the commands `ommodmn`, `ommodu`, `ommod-pdl`, `ommodent`, and `omldapmodify` generate Internet addresses automatically, when automatic Internet address mapping is in operation. The default is FALSE.
INET_DISPLAY_IA_COMMENTS= TRUE	Determines whether the POP3 and IMAP4 interfaces display the comment, or display name, part of an Internet address in a message. The default is TRUE. Set the option to FALSE to prevent the comment part of the Internet address from being displayed.
INET_INLINE_FILE_MAX_SIZE= *bytes*	Determines which body parts of MIME messages generated by the Internet Mail Gateway are inline and which are attachments. Body parts whose size is greater than the value of this option will be attachments, while other body parts will be inline. Set a value of 0 to cause all body parts to be inline. Set a value of -1 to cause all body parts to be attachments.
INET_INLINE_FNAME_ALLOWED= FALSE	Determines whether MIME messages generated by the Internet Gateway or prepared for browsing by POP3 or IMAP4 clients can have `filename=` in inline body part `Content-Disposition` lines. If the option is set to FALSE (the default), Inline body parts cannot have `filename=` in the `Content-Disposition` line even if a candidate filename exists. Set this option to TRUE to allow inline body parts to have `filename=` in the `Content-Disposition` line, if a candidate filename has been selected.
INET_NO_IA_IN_ORN=FALSE	Determines whether the incoming Internet Mail Gateway saves the Internet address of the sender, each recipient and DL member in the Scalix message. When set to FALSE (the default), the addresses are saved in the message. Note that this option applies to the names of Internet mail users and not to the names of Scalix users.
INET_NO_IA_COMMENTS=FALSE	Determines whether comments present in Internet addresses are included without alteration in outgoing messages. The default is FALSE, causing such comments to be included.
INET_USE_AUTO_IAM=TRUE	Specifies whether Internet addresses will be automatically created when configuring users, and -mapped at the Internet Mail Gateway and the POP3 and IMAP interfaces.

Internet Mail Gateway Options

Option	Description
BRW_INLINE_PARTS=1	Specifies the number of parts in a multi-part item that are marked as inline if there is no other information about the part. The default is 1.
BRW_ITEMSUB_IS_FNAME=TRUE	For those items which do not have an Original Filename specified, determines whether the Subject of the item will be used to generate a filename for the item. Leave this option at its default (TRUE) if you wish to use the item Subject as the filename (provided the value of BRW_T61_ITEMSUB_IS_FNAME has not caused a filename to be generated). Encoding is determined by BRW_MIME_FNAME_ENCODING.Note that if BRW_MIME_FNAME_ENCODING=D then this takes precedence over BRW_ITEMSUB_IS_FNAME. Set the option to FALSE to prevent the Subject from being used to generate the filename (although a filename can still be generated using one of the other fields).
BRW_MIME_EXPLICIT_ASCII= FALSE	Specifies the character set to be US-ASCII for plain text content types. The MIME standards specify the default character set to be US-ASCII for text/plain content types, and you should therefore leave this at its default (FALSE) unless your client cannot display such content unless the character set is explicitly defined.
BRW_MIME_FNAME_ENCODING=Q	Specifies the method for encoding MIME names and filenames used in the POP3 and IMAP4 interfaces. The possible values are: D: forces outgoing non-text filename to meet DOS filename conventions N: no encoding Q: quoted-printable encoding; this is the default B: base64 encoding
BRW_MIME_OMIT_DEF_CTENC_ HDR=F or N	If set toT for TRUE or Y for YES, the Content-Encoding header is omitted if it is the default 7 bit. The default is F or N.
BRW_MIME_SPACE_OK_IN_FNAME=TR UE	Specifies that spaces are allowed in filenames based on the T.61 subject of a body part.
BRW_MIME_SUBJ_BENC_ NONASCII=F	When BRW_MIME_SUBJECT_ENCODING is set to B for base64 encoding, you can set this option (using either T for TRUE or Y for YES), to encode only non-ASCII characters in MIME subjects using base64. The default is F or N.
BRW_MIME_SUBJ_NO_SPACE_ SEPS=FALSE	If set (using TRUE or YES), a space separator between encoded and non-encoded data is not generated. This option can only be set when BRW_MIME_SUBJECT_ENCODING=B and BRW_MIME_SUBJ_BENC_NONASCII=T. The default is FALSE or NO. Note that setting this option generates messages in a form that is not strictly compatible with RFC1522.

Option	Description
BRW_MIME_SUBJECT_CHARSET= NULL	Specifies the POP3 and IMAP4 MIME subject character set when it is different from the body part text. This option is for RFC 1557, which uses ISO-2022-KR in the body part text and EUC-KR (equivalent to KSC5601) in the subject.
BRW_MIME_SUBJECT_ENCODING= Q, B or N	Specifies the method for encoding the message subjects of MIME messages retrieved from the POP3 and IMAP4 Servers. The methods available are: D: forces outgoing non-text filename to meet DOS filename conventions N: no encoding Q: quoted-printable encoding; this is the default B: base64 encoding
BRW_MIME_SUBJECT_FOLDING= F, or N	If set to T or Y, folds subject headers according to RFC 1522 rules (at 76 bytes after encoding). Multibyte characters are sometimes folded at slightly less, to avoid splitting characters and to handle escape sequences correctly. The default is F or N.
BRW_MIME_TEXTFILE_ENCODING =Q, B, N, or ?	Specifies the method for encoding the message texts of MIME messages retrieved from the POP3 and IMAP4 Servers. The methods available are: ?: use the relevant mapping in the mime.types file (the default) N: no encoding Q: quoted-printable encoding; this is the default B: base64 encoding
BRW_NAME_MAPPING=FALSE	If set, the originator's name and address will be mapped to the keyed INTERNET-ADDR attribute (number 167) in the Directory entry for that user. The Directory entry must contain the user name and domain name in the format expected by Sendmail. Routing set up within Scalix and Sendmail must correspond to the addresses used in mappings. Note that mappings occur only when there is an exact match between the name and address in the message and the Directory entry attribute INTERNET-ADDR. You can specify which Directory to use for name/address mappings using the UX_NAME_MAPPING_DIR option. You can specify a Directory entry attribute to use other than INTERNET-ADDR using the UX_NAME_MAPPING_ATTRIB option. The default is FALSE.

Option	Description
BRW_NO_RETAIN_IF_CONVERTED=FALSE	Determines if a message containing an alternative filetype read through the POP3 or IMAP4 Server retains the original format file along with the converted plain text version. By default, this option is set to FALSE; alternative filetypes are retained along with the converted plain text version. If this option is set to TRUE, the alternative file is discarded after it has been converted into a text file. The resultant MIME message is created using just the converted text file. *filetype* is a filetype (or a comma-separated list of filetypes) configured in Scalix to be discarded. For example, to discard an original RTF file after conversion to plain text, set this option to BRW_NO_RETAIN_IF_CONVERTED=2130.
BRW_T61_ITEMSUB_IS_FNAME=F	If set to T, the T61 item subject is used for the filename when browsing POP3 and IMAP4 mail messages. The encoding is determined by the setting of BRW_MIME_FNAME_ENCODING; if BRW_MIME_FNAME_ENCODING is set to D, it takes precedence over this option. The default is F.
INET_USE_X400_ATTS_FOR_LOOKUP=TRUE	Determines whether name mapping, using Directory lookup, at the outgoing Internet Mail Gateway uses only X.400 attributes. When this option is TRUE (the default), the Outgoing Internet Mail Gateway will only use X.400 attributes when it performs Directory lookup to map OR addresses to Internet addresses.
MAX_MIME_BROWSERS=25	Specifies the maximum number of MIME browsers that the MIME Browser Controller can have in its pool. The default is 25. Specify a higher value to provide a faster response to IMAP4 and POP3 connection requests. Specify a lower value to conserve system resources.
MIME_CACHE_TARGET_SIZE=1	Specifies the target size in megabytes of the mime cache (~Scalix/temp/mime_cache). The cache can grow larger than this if everything in the cache is being used, but unused items will be deleted to keep the size under control. The default is 1.
MIN_MIME_BROWSERS=0	Specifies the minimum number of MIME browsers that the MIME Browser Controller can have in its pool. The default is 0. When the Mime Browser Controller starts, it will start the number of browsers specified by this option.
UX_MIME_SUBJECT_CHARSET=*client_character_set*	Specifies the character set to be used for the incoming and outgoing MIME subject, when it is different from the body part text. client-character-set is the name of any client character set configured in Scalix; for example, KSC5601. This option is for RFC-1557, which uses ISO-2022-KR in the body part text and EUC-KR (equivalent to KSC5601) in the subject.
UX_NAME_MAPPING_ATTRIB=*attribute_tag*	Specifies a Directory entry attribute to use other than INTERNET-ADDR. *attribute_tag* is the internal (numeric) form of the Scalix attribute.

Option	Description
UX_NAME_MAPPING_DIR= *Directory_name*	If set, the Directory you specify here is used to map the originator or recipient's name and address to an Internet address specified in the `INTERNET-ADDR` attribute of the user's Directory entry, as the message passes through the Internet Mail Gateway. The Directory must be a shared Directory. If this option is not set, the default system Directory is used. See also the `UX_NAME_MAPPING_DIR_PASSWD` option.
UX_NAME_MAPPING_DIR_PASSWD =*password*	If you set the `UX_NAME_MAPPING_DIR` option, you can specify a Directory password using this option.
UX_NO_ROUTE_CHECK=TRUE	If set, the Internet Mail Gateway does not check O/R Addresses in the incoming messages. Normally, the Internet Mail Gateway checks if valid routes are configured for O/R Addresses in the ARPA heading information of each incoming message.
UX_PRE_5_20_COMPATIBILITY_ MODE=TRUE	When set to `TRUE` (the default), this option causes Scalix to create a `WINMAIL.DAT` attachment for outgoing messages that contain MAPI properties, and to not decode `WINMAIL.DAT` attachments for incoming messages.
UX_USE_ARPA_SENDER=TRUE	If set, the incoming Internet Mail Gateway constructs the Scalix "From" address of incoming messages from the value of the Sender: token in the ARPA header rather than from the SMTP `Mail From` command.
UXI_AUTO_REPLY_BULK_MAIL= FALSE	Specifies whether the incoming Internet Mail Gateway should allow auto-replies to bulk mailing list messages. Bulk mailing list messages are those that contain one of the following lines in the ARPA header: Precedence: bulk Precedence: list Precedence: junk By default (this option set to `FALSE`), bulk mailing list messages are treated in a very similar way to auto-forwarded messages, and do not allow auto-replies. Set this option to `TRUE` to allow auto-replies to bulk mailing list messages.
UXI_DDATYPE_HPMEXT=TRUE	If set, then messages coming in through the Internet Mail Gateway that have their internet mail addresses copied into the DDA of the Scalix addresses will use a DDA with a type of `HPMEXT1` rather than the default `RFC-822`.
UXI_DOS_FNAME=TRUE	Specifies that Scalix message subjects for non-text attachments in MIME messages sent to Scalix, are used to create DOS filenames.

Option	Description
UXI_DO_1327_SENDER_MAP=TRUE	When set, the ARPA Sender of an Internet message will be shown as the Creator of the message. Any replies to this message will therefore be sent to the Sender. This is the behavior described in RFC 1327. Note that this can not be the required behavior; for example, it means that Replies to a mailing list can be directed to the Creator, rather than to the subscribers.
UXI_KEEP_ARPA_ADDRESS=TRUE	If set, the Internet Mail Gateway will preserve the ARPA address of an incoming message even if the header contains an ARPA-encoded Scalix address. Normally, the Internet Mail Gateway checks the ARPA heading information of each incoming message and, if there is an ARPA encoding of a Scalix address, the ARPA address is discarded and the Scalix address used instead.
UXI_KEEP_MIME_ARPA_HEADER=TRUE	If set, the Internet Mail Gateway will include the ARPA header of an incoming MIME message in the Scalix message.
UXI_KEEP_UUENC_ARPA_HEADER=TRUE	If set, the Internet Mail Gateway will include the ARPA header of an incoming UUENCODE message in the Scalix message.
UXI_MIME_CS_AUTODETECT	Scalix scans text MIME body parts which are marked as being in an ASCII or ISO8859_1 character set to check whether they have been incorrectly labeled and are actually another character set type that Scalix can recognize with a higher degree of certainty. This scan is enabled by default by setting UXI_MIME_CS_AUTODETECT=FALSE, but can be disabled to increase Internet Gateway performance.
UXI_NAME_MAPPING=TRUE	If set, the originator's name and address will be mapped to the keyed INTERNET-ADDR attribute (number 167) in the Directory entry for that user. The Directory entry must contain the user name and domain name in the appropriate format. Routing set up within Scalix and Sendmail must correspond to the addresses used in mappings. Note that mappings occur only when there is an exact match between the name and address in the message and the Directory entry attribute INTERNET-ADDR. You can specify which Directory to use for name/address mappings using the UX_NAME_MAPPING_DIR option. You can specify a Directory entry attribute to use other than INTERNET-ADDR using the UX_NAME_MAPPING_ATTRIB option.
UXI_NO_CONVERT_REPORTS=FALSE	When set to FALSE, Internet acknowledgments and acknowledgment requests are converted to their Scalix equivalents. If you set this option to TRUE, they are not converted, but are passed into Scalix as Messages.
UXI_NO_INET_OBJFILES=FALSE	When set to FALSE, the contents of Internet headers are preserved in object files. Certain clients, such as the IMAP4 and POP3 clients, can make use of these object files. If set to TRUE, Scalix will not create these object files, and this header information will therefore be lost.

Option	Description
UXI_NO_UUDECODE_STRING= text string	If set, UUENCODEd parts of messages that are not of a recognized format, such as MIME or RFC1154, will not be decoded if the message body contains a text string that matches the one you specify here. If this option is not set, or no text string is supplied for it, such messages will have their UUENCODEd parts decoded into separate binary attachments.
UXI_PASSIVE_RECIPS_MAPI_ ENABLED=FALSE	Determines whether passive recipients (that is, those recipients for which the Internet Mail Gateway does not have responsibility) appear to Outlook users with the "Send In RTF" flag set. When this option is set to FALSE (the default), then such recipients are assumed not to be MAPI-enabled, and the "Send In RTF" flag is not set. Set this option to TRUE if you wish to have the "Send In RTF" flag set for these recipients.
UXI_PRESERVE_MAPI_MSG_CLASS=FA LSE	Specifies whether the MAPI message class of certain incoming messages is converted. To interoperate with Exchange, the Internet Mail Gateway must convert the MAPI message class of certain messages received from the Exchange Internet Mail connector for the Scalix MAPI Service Providers. The default is FALSE. See also the UXO_PRESERVE_MAPI_MSG_CLASS option.
UXI_SUPPRESS_ARPA_HEADER= TRUE	Suppresses, at a system level, the generation of the ARPA header for incoming messages from the Internet Mail Gateway. Note that MIME-encoded messages have ARPA headers suppressed by default. To enable MIME-encoded messages to have ARPA headers, you must set UXI_KEEP_MIME_ARPA_HEADER=TRUE.
UXI_TREAT_AS_MIME_SUBJECT= T or Y	If set (using either T for TRUE or Y for YES), incoming messages from the Internet Mail Gateway's UUENCODE or SHAR route (in other words, messages in which the ARPA headers did not contain a Mime-Version: 1.0 tag) but which have MIME-conformant subjects, have their subjects decoded as if they came via the MIME route, and are subject to all other settings for MIME subjects.
UXI_UNIX_MAIL_CHARSET= *character_set*	Specifies the character set used in messages coming in through the Internet Mail Gateway. Only use this option to specify non-Latin character sets. For Latin character sets, Scalix assumes the same character set is being used as was used for outgoing messages; that is, the character set to which ISO8859/1 text was converted to in the unix-out.str or mimeout.str file. If the UXI_UNIX_MAIL_CHARSET option is set, the file ~/sys/unixin.str or mimein.str can be used to specify conversions from this character set to a suitable interchange character set.

Option	Description
UXI_UUDECODE_ARPA_TOKEN=*string*	Specifies the token which, if present in the ARPA header of an incoming internet mail message, results in any UUENCODEd parts in the message being decoded. The default is to decode UUENCODEd parts in all messages. To prevent any messages containing UUENCODEd message parts from being decoded (except those that conform to RFC 1154), specify a null ("") string.
UXO_ADD_DELIM=TRUE	Specifies that a leading / is inserted in front of an O/R Address that is being used within internet mail. The / is inserted only if the O/R Address is format 2 (attribute format) and is enclosed in inverted commas. (This is done when attributes in the O/R Address contain characters that have a special meaning to internet mail.) Inserting the / ensures that Sendmail identifies the message as a message for Scalix.
UXO_CHECK_TYPES_OF_DDA=*DDA_type*	Specifies the DDA types that are acceptable as valid internet addresses, to enable the UAL Client Interface and the Internet Mail Gateway to route the message to the correct destination for the recipient. *DDA_type* is one of the following: One or more valid DDA types, for example: RFC-822 HPMEXT1 HPMEXT2 HPMEXT3 HPMEXT4 You can specify up to 10 DDA types; if you do specify more than one type, separate them with commas. For example: RFC-822,HPMEXT1,HPMEXT2,HPMEXT3,HPMEXT4 FALSE No DDA type checking is performed. This behavior is as for Scalix systems before B.05.10. If your Scalix Directory contains DDAs with no type specified and they are not valid internet addresses, you are recommended to set this option to: UXO_CHECK_TYPES_OF_DDA=RFC-822,HPMEXT1, HPMEXT2,HPMEXT3,HPMEXT4 If this option is not present, the default setting is: UXO_CHECK_TYPES_OF_DDA=,RFC-822,HPMEXT1, HPMEXT2,HPMEXT3,HPMEXT4 The leading comma means that DDAs with no type specified are also acceptable as valid internet addresses.

Option	Description
UXO_ITEMSUB_IS_FNAME=FALSE	If set to TRUE, the item subject is used for the filename in outgoing MIME Internet Mail messages, if no Original filename is present and if UXO_T61_ITEMSUB_IS_FNAME has not caused a filename to be generated. The encoding is determined by the setting for UXO_MIME_FNAME_ENCODING; if UXO_MIME_FNAME_ENCODING is set to D, it takes precedence over this option. The default is FALSE; this setting prevents a filename from being generated from the item subject for the Content-Disposition header. See also: UXO_MIME_FNAME_ENCODING UXO_T61_ITEMSUB_IS_FNAME INET_INLINE_FNAME_ALLOWED
UXO_MIME_FNAME_ENCODING=D	Specifies how MIME names and filenames are encoded at the outgoing Internet Mail Gateway. The options are: D: forces outgoing non-text filename to meet DOS filename conventions N: no encoding Q: quoted-printable encoding; this is the default B: base64 encoding
UXO_MIME_OMIT_DEF_CTENC_HDR=T or Y	If set (using either T for TRUE or Y for YES), the Content-Encoding header is omitted if it is the default 7 bit.
UXO_MIME_SPACE_OK_IN_FNAME=TRUE	Specifies that spaces are allowed in filenames based on the T.61 subject of a body part.
UXO_MIME_SUBJ_NO_SPACE_SEPS=TRUE	If set (using TRUE or YES), a space separator between encoded and non-encoded data is not generated. This option can only be set when UXO_MIME_SUBJECT_ENCODING=B and UXO_MIME_SUBJ_BENC_NONASCII=T. Note that setting this option generates messages in a form that is not strictly compatible with RFC1522.
UXO_MIME_SUBJECT_BENC_NONASCII=T	When UXO_MIME_SUBJECT_ENCODING is set to B for base64 encoding, you can set this option (using either T for TRUE or Y for YES), to encode only non-ASCII characters in MIME subjects using base64.
UXO_MIME_SUBJECT_ENCODING=Q, B or N	Specifies the method for encoding MIME subjects of outgoing messages. The methods available are: N: no encoding Q: quoted-printable encoding; this is the default B: base64 encoding If UXO_MIME_FNAME_ENCODING is not set, this option is used for filename encoding, as well.

Option	Description
UXO_MIME_SUBJECT_FOLDING= T, or Y	Folds subject headers according to RFC 1522 rules (at 76 bytes after encoding). Multibyte characters are sometimes folded at slightly less, to avoid splitting characters and to handle escape sequences correctly. Enter either T for TRUE or Y for YES to set this option.
UXO_MIME_TEXTFILE_ENCODING=Q, B or N	Specifies the method for encoding message text of outgoing MIME messages. The methods available are: N: no encoding Q: quoted-printable encoding; this is the default B: base64 encoding
UXO_NAME_MAPPING=TRUE	If set, the recipient's name and address will be mapped to the keyed INTERNET-ADDR attribute (number 167) in the Directory entry for that user. The Directory entry must contain the user name and domain name in the format expected by Sendmail. Routing set up within Scalix and Sendmail must correspond to the addresses used in mappings. If the recipient already has an Internet mail name and address configured in the DDA fields in the message or in the entry retrieved from the Directory, this will be used in preference to the INTERNET_ADDR attribute value. You can specify which Directory to use for name/address mappings using the UX_NAME_MAPPING_DIR option. You can specify a Directory entry attribute to use other than INTERNET-ADDR using the UX_NAME_MAPPING_ATTRIB option.
UXO_NO_RETAIN_IF_CONVERTED=FALSE or *filetype*	Determines if a message containing an alternative filetype sent through the Internet Mail Gateway retains the original format file along with the converted plain text version. By default, this option is set to FALSE; alternative filetypes are retained along with the converted plain text version. If this option is set to TRUE, the alternative file is discarded after it has been converted into a text file. The resultant MIME message is created using just the converted text file. *filetype* is a filetype (or a comma-separated list of filetypes) configured in Scalix to be discarded. For example, to discard an original RTF file after conversion to plain text, this option is set to UXO_NO_RETAIN_IF_CONVERTED=2130.
UXO_PRESERVE_MAPI_MSG_CLASS=FALSE	Specifies whether the MAPI message class of certain outgoing messages is converted. To interoperate with Exchange, the Internet Mail Gateway must convert the MAPI message class of certain messages destined for the Exchange Internet Mail connector for the Scalix MAPI Service Providers. The default is FALSE. See also the UXI_PRESERVE_MAPI_MSG_CLASS option.
UXO_SHAR_ARGS=*arguments*	Specifies the arguments used by the shar program when it is started by the Internet Mail Gateway. The default arguments are -bc. See also UXO_SHAR_COMMAND.

Option	Description
UXO_SHAR_COMMAND=*command*	Specifies the program used by the Internet Mail Gateway to create a shell archive package. The default is `shar`. See also UXO_SHAR_ARGS.
UXO_T61_ITEMSUB_IS_FNAME=T	If set, the T61 item subject is used for the filename in outgoing MIME Internet Mail messages. The encoding is determined by the setting for UXO_MIME_FNAME_ENCODING; if UXO_MIME_FNAME_ENCODING is set to D, it takes precedence over this option.
UXO_TREAT_AS_MIME_SUBJECT= T or Y	If set (using either T for TRUE or Y for YES), messages going out via the Internet Mail Gateway's UUENCODE or SHAR route that have MIME-conformant subjects will have their subjects encoded as if going out via the MIME route, and will be subject to all other settings for MIME subjects.
UXO_USE_SENDER_DDA=TRUE	Specifies whether a Domain Defined Attribute (DDA) is used directly when mapping the sender address in the outgoing Internet Mail Gateway. If set to TRUE, the Internet Mail Gateway maps DDAs in the sender address in the same way as it does for DDAs in the recipient address; that is, any internet address specified in the DDA is used directly. If this option is not present or is set to FALSE, the Internet Mail Gateway does not use the DDA directly when mapping the Internet address of the sender.

Item Structure Server Options

Option	Description
ISL_DISABLE_LOGGING=TRUE	Disables logging by the Item Structure Server, of structural changes made to the Message Store. This option is set when the Item Structure Database is not required, in order to save disk space. The option takes precedence over ISL_LOG_IF_OFF=TRUE.
ISL_LOG_IF_OFF=TRUE	Enables logging of structural changes made to the Message Store, when the Item Structure Server daemon is not running. Logging is performed directly to the Item Structure Server log files, which reduces performance. The option ISL_DISABLE_LOGGING=TRUE takes precedence over this option.

Local Delivery Service Options

Option	Description
LD_ADD_ACKS_AS_TO	If set, the address returned in an acknowledgment that cannot be matched in the original distribution list of the message in the Outbox is added to the distribution list as a "To" record rather than as a "New Recipient" record.
LD_AUTOREPLY_CHECK_ON=TRUE	Each time a user configures auto-reply, Scalix creates a text file in the user's `/g` directory under `~/user/` which contains a list of addresses to which automatic replies have been sent since the current auto-reply session was created. With this option set, Local Delivery checks users' address list files against the address in each received message's transaction file. If a match is found, an automatic reply is not generated. This prevents more than one automatic reply from being generated for each unique sender address.
LD_AUTOREPLY_EXPIRY_TIME=*no_of_days*	Specifies the number of days an address can be present in the auto-reply address list file before it is removed.
LD_MAX_NEST_LEVEL=*depth*	Specifies the maximum level of nesting allowed in a message before further nested parts are flattened by the Local Delivery Service. A value of `0` means that all delivered messages will be flattened. See `SR_MAX_NEST_LEVEL` for more information.
LD_READ_ACK_ON_AUTOPRINT=FALSE	If set, when a message is automatically printed, no "read" acknowledgment is returned to the originator. The default is to return a "read" acknowledgment when a message is automatically printed.
LD_TRACE_DISP_ACT=SHOW_ADMIN	Shows trace information on all messages received by anyone with Scalix administration permissions.

LDAP Server Options

Option	Description
LDAP_MB_CN_IS_GS_IN_FILTER=FALSE	This option only has effect when the LDAP session is multibyte, and you have not created an explicit Scalix attribute COMMONNAME. When the LDAP client sends a search filter that contains the LDAP COMMONNAME attribute, the LDAP Server uses a built-in parsing method to determine how to convert this to Scalix attributes. By default, it assumes that the COMMONNAME attribute contains the Scalix attributes SURNAME, GIVENNAME in that order. If your LDAP clients construct COMMONNAMES in the reverse order, you must set this option to TRUE. For example, if your LDAP client uses "Japanese-surname Japanese-givenname" as the COMMONNAME, leave this option at its default. The LDAP Server will correctly interpret this as GIVENNAME=Japanese-givenname and SURNAME=Japanese-surname. However, if your LDAP client uses Japanese-givenname Japanese-surname as the COMMONNAME, the LDAP Server will interpret this as GIVENNAME=Japanese-surname and SURNAME=Japanese-givenname unless you set this option to TRUE.
LDAP_MB_CN_IS_GS_IN_SYNTH_OUT=FALSE	This option only has effect when the LDAP session is multibyte, and you have not created an explicit Scalix attribute COMMONNAME. When the LDAP Server synthesizes the LDAP COMMONNAME attribute using the Scalix attributes SURNAME and GIVENNAME, it puts them in the order SURNAME, GIVENNAME. If you want COMMONNAMES to appear in the reverse order, you must set this option to TRUE. For example, when this option is left at its default value, the name "Japanese-givenname Japanese-surname" will be returned from the LDAP Server as Japanese-surname Japanese-givenname. However, if you set this option to TRUE, the name will be returned as Japanese-givenname Japanese-surname.
LDAP_SEQUENTIAL_SEARCH=""	If you do not set this option, the LDAP Server will not, in general, issue sequential searches of the Scalix directories. Instead, it will search using the indexes of keyed attributes, to keep search time to a minimum. However, it will issue sequential searches under certain circumstances, such as when the DA_IGNORE_INDEXES option is set to TRUE. Set this option to TRUE if you want the LDAP Server to issue sequential searches. This enables you to search for attributes that are not keyed, but searches could take a long time. Set this option to FALSE to prevent the LDAP Server from ever issuing sequential searches. This will keep search times to a minimum, but can prevent the LDAP Server from finding all entries that match a given filter.

Option	Description
OMLDAP_REMOVE_LEADING_ WILDCARDS=TRUE	If present and set to TRUE, leading wildcard characters (*) are stripped from substring filters when the LDAP Server searches a Scalix Directory for entries that match criteria specified by a search filter. This option causes filters of the form "(cn=*name**)" to be converted to the form "(cn=*name**)". That is, the LDAP Server matches the filter "(cn=*name**)" to all entries in the underlying Scalix Directory whose SURNAME or COMMON-NAME attributes start with name. This causes fewer system resources to be used when searching. If not present or set to FALSE, the leading wildcards are not stripped, and the LDAP Server searches for all SURNAME or COMMON-NAME attributes containing name. For example, if OMLDAP_REMOVE_LEADING_WILDCARDS is set to TRUE, "(cn=Marion Brand*)" would be matched to all entries whose SURNAME starts with "Brand" or whose COMMON-NAME starts with "Marion Brand". If this option is set to FALSE, "Ann-Marion Brandson" would be considered a match.

Non-Delivery Notification Options

Option	Description
NDN_EM_SERIOUS_ONLY=TRUE	Sends Non-Delivery Reports for serious errors to the Error Manager only. Sends Non-Delivery Reports for simple addressing problems to the originator only. If this option is not set, by default Non-Delivery Reports for simple addressing problems are sent to both the Error Manager and the originator.
NDN_NO_ALTERNATES=TRUE	If this option is set, alternate names are not placed in a Non-Delivery Notification if the original message contained an ambiguous O/R Name.

Notification Server Options

Option	Description
NS_INITIAL_MEM=*bytes*	Specifies the initial memory size of the Notification Server. Use this option to increase the initial memory size from 65536 (the default). This value is suitable for up to approximately 1200 configured and active users. You might want to increase this value if a larger number of users are configured such that, just after startup, the shared memory segment is repeatedly enlarged.

Offline Folder Synchronization Options (Outlook Clients)

Option	Description
OFS_ENABLED=FALSE	Specifies whether folder synchronization is enabled on the Scalix Server. The default is TRUE. If this option is set to TRUE, it can be overridden on a per-user basis by setting it to FALSE in the relevant user-specific configuration files.
OFS_LOG_AGE_LIMIT=*days*	When the age of a change log entry exceeds this value, it can be deleted when the change log file is compacted. Use this option to prevent change logs from growing too large when logging is activated but synchronization is not taking place for any reason. Take care when you set this value, since removal of any valid entries will cause the entire folder to be resynchronized. A value you set in `general.cfg` can be overridden on a per-user basis by setting a lower value for this option in the relevant user-specific configuration file.
OFS_LOG_SIZE_LIMIT=*kilobytes*	Specifies, in kilobytes, the maximum size of the folder synchronization change log. Set a value between 20 and 10,000 KB. The default is 100 KB. When the size of a change log exceeds this value, the older entries can be deleted when the change log file is compacted. Use this option to prevent change logs from growing too large when logging is activated but synchronization is not taking place for any reason. Take care when you set this value, since removal of any valid entries will cause the entire folder to be resynchronized. A value you set in `general.cfg` can be overridden on a per-user basis by setting a lower value for this option in the relevant user-specific configuration file.
OFS_WORK_FILE_DIR=*temp-directory*	Specifies the location of the temporary files created during folder synchronization. Normally (when this option is not set), these temporary files are stored in `~/temp`. Specify a value for this option to cause all these files to be written to a different location. This allows you to use a high-speed (and possibly low-recovery) file system (for example, a RAM disk) to store these temporary files. The directory you specify must have: permissions 771 a group of `scalix` an owner of `scalix` a path length of 225 characters or less For example, to create a directory called `temp-ofs`, enter the following commands: mkdir $(omrealpath '~/temp-ofs') chown scalix:scalix $(omrealpath '~/temp-ofs') chmod 771 $(omrealpath '~/temp-ofs)

Omscan Options

Option	Description
GS_DONT_SPLIT_FC=FALSE	When set to TRUE, `omscan` reports a single value for the size of a user's filing cabinet and waste basket combined, instead of separate figures for the filing cabinet and the waste basket.
SCN_KEEP_DATA_ORPHANS=FALSE	If set, files reported as orphans by `omscan` are not moved to the directory `~/orphans` but are deleted.
SCN_ORPHAN_DELETE_DELAY=*number_of_days*	Sets the age limit for orphan files before they are reported by `omscan` and moved to the directory `~/orphans`. The default is 1 day.
SCN_PREV_ORPHAN_DELETE_DELAY=*number_of_days*	The number of days before a file in the directory `~/orphans` is deleted by the next run of `omscan -d`. The default is 30 days.
SCN_TEMP_DELETE_DELAY=*number_of_days*	Sets the age limit for files in `~/temp` before they are deleted by `omscan`. The default is 7 days.
SCN_TMP_DELETE_DELAY=*number_of_days*	Sets the age limit for files in `~/tmp` before they are deleted by `omscan`. the default is 7 days.

POP Server Options

Option	Description
POP3_IGNORE_SERVERNAME=FALSE	Determines whether the POP3 Server uses the characters following the @ character in a username as the Server name for this user. When set to FALSE (the default), the name part of the username (up to and including the @ character) is stripped off and the remainder is used as the Server name to which the POP3 connection is relayed. Set this option to TRUE to prevent the POP3 connection being relayed to another Server.
POP3_MAILSTORE_HOST=*hostname*	Specifies the fully qualified domain name of the Scalix host to which the POP Server connects; for example, `omsrv1.acme.company.com`. Use this option when the POP Server does not reside on the same machine as the Scalix system that contains the relevant message store.
POP3_MAX_THREADS=*integer*	Restricts the number of threads that a single `pop3.server` process will use. By default, the value used is the maximum allowed from system resources (including system thread limits as well as limits in available file descriptors). Specify a value here to limit the number of threads used to a value less than the default. If you specify a value higher than the default, it will have no effect.

Option	Description
POP3_RECORD_EMPTY_SIGNON= FALSE	Determines whether POP3 user signons are recorded for empty In Trays. When a user with items in their In Tray signs on using POP3, the signon is recorded, and the last signon time for the user is updated. This causes the signon to take longer than if the signon is not recorded. When this option is FALSE (the default), if a user with an empty In Tray signs on using POP3, the signon is not recorded, and is faster. Set this option to TRUE to cause user signons to be recorded even when the user's In Tray is empty. This causes slower signons, but allows you to discover the last signon time for the user or to use `omstat -u` and `omstat -s` to report on POP3 users.

Public Folder Server Options

Option	Description
BBS_ALLOW_LOCAL_SYNC=FALSE	Specifies whether public folder synchronization can occur between public folders on the same machine. By default, this option is `false`, and an attempt to synchronize public folders on the same machine results in a `warning` message in the log file.
BBS_CUST_CHECK_TIME=*minutes*	Sets the amount of time a public folder server spends in "import" mode before checking if any of its synchronization timers have expired. the default is 5 minutes. A public folder server toggles between import and export mode. It stays in import mode for the specified amount of time before checking the synchronization timers, and then if the timers have expired, it changes to export mode. when it has done all the exports, it changes back to import mode.
BBS_DELETE_MASTER_BY_SYNC= FALSE	Specifies that deletion of a Slave item from a Public Folder will not be propagated to its Master item. Set this option to TRUE if you want deletion of a Slave item to result in deletion of its Master item. (BBS_PROPAGATE_SLAVE_DELETION must also be set to TRUE for this to happen.)
BBS_DELLOG_RETENTION_PERIOD=*hours*	Specifies the Retention Period, in hours, for delete log files. The default is 24.
BBS_PROPAGATE SLAVE DELETION=FALSE	Specifies that deletion of a Slave item from a Public Folder will not be propagated to any other Slave copies or to the Master item. Set this option to TRUE if you want deletion of a Slave item to result in deletion of all other Slave copies of this item. (BBS_DELETE_MASTER_BY_SYNC must also be set to TRUE if you want deletion of a Slave item to result in deletion of its Master.)

Option	Description
BBS_SEND_OBJECT_FILES=TRUE	Specifies whether object files attached to messages or items are included when the message or item is exchanged during the Public Folder synchronization process. If set to TRUE, any object files attached to messages or items are included when the message is sent to another Public Folder Server. Note that this option affects only object files attached to messages or basic items on the Public Folder; object files attached directly to Public Folders are not synchronized.
BBS_SYNC_MESG_PRIORITY=*priority*	Specifies the priority with which Public Folder synchronization messages are sent. If Public Folder synchronization messages are slowing down other Scalix operations, you might want to use a lower priority for such messages. Alternatively, if Public Folder synchronization messages are taking too long, you might want to use a higher priority. Set the priority value to one of 0 (normal), 1 (nonurgent), or 2 (urgent).

Queue Options

Option	Description
QM_DONT_READ_MSG_AT_START=FALSE	When set to TRUE, this option specifies that all messages currently in the Scalix queues remain stationary when Scalix is restarted. New messages will be processed normally. Set this option to TRUE if a major problem is encountered when the queue manager attempts to read queued messages from disk when Scalix starts up. This allows Scalix to start. Set the option back to FALSE and restart Scalix when the problem is resolved.
QM_FAILURE_DELAY_SEC=*seconds*	Specifies the number of seconds between messages being retried by the queue manager. When a message fails to be processed because the process that was handling the message died, the queue manager delays the message for this number of seconds before retrying the message. The default is 30. See also QM_MAX_FAILURES.

Option	Description
QM_MAX_FAILURES=*integer*	Sets the number of times the queue manager will attempt to retry a message before giving up and putting the message on the POISON queue. A failure occurs when the process which received the message dies before informing the queue manager that it has successfully dealt with the message. A value between 1 and 4 is normally suitable. A value of 0 will cause the queue manager to place the message on the POISON queue immediately on failure (that is, no retries). Too large a value can cause services to abort repeatedly. See also QM_FAILURE_DELAY_SEC.
Q_TIME_OUT=*seconds*	Sets the amount of time processes attempting to read a request from a Scalix queue will wait before timing-out. Setting this value low will ensure that processes remain swapped-in. The default is 30 seconds.

Search Server Options

Option	Description
SE_DEFAULT_DELAY= *number_of_seconds*	Specifies the delay in seconds between runs of a persistent (that is, automatically repeated) background search of the Message Store. The Search Server checks for a specified delay in the following sequence: If a delay is specified in the search request transaction file, that delay is used. If no delay is specified in the search request transaction file (or it is set to zero), the delay specified in this SE_DEFAULT_DELAY option is used. If no delay is specified either in the search request transaction file or the SE_DEFAULT_DELAY option, a default delay of 300 seconds is used.
SE_MAX_CHILDREN= *max_number_of_child_processes*	Specifies the maximum number of child search processes that the Search Server can create. Each child process can execute only one search at a time. This option limits the number of background searches that can be performed simultaneously, but not the number of background searches that can be queued. The default number of child processes is 20.
SE_MAX_OVERDUE_TIME= *number_of_seconds*	Specifies the time after which an overdue persistent background search takes priority over one-off searches. The Search Server normally gives priority to one-off searches. When the time specified in this option is reached, the Search Server gives a persistent search priority over the one-off searches. This prevents persistent searches from being permanently blocked by a long queue of one-off searches. The default time is 300 seconds.

Service Router Options

Option	Description
NDN_MSGFLAGS_OVERRIDE_RULE_ACTION_RETURN=FALSE	By default, messages created as a result of the RETURN action for a message delivery ruleset has the original message attached, even if the original message has a flag specifying that contents must not be included in non-delivery notifications. If you set NDN_MSGFLAGS_OVERRIDE_RULE_ACTION_RETURN to TRUE and the return of contents is not requested, the original message is not attached.
RSL_BLANK_SUBJECT_BS_CHAR=FALSE	If the subject mapper is a shell script, a message subject containing a backslash () causes problems as the script interprets these as escape characters. The Service Router and Deferred Mail Manager replace any backslashes with an empty space. If the subject mapper is not a shell script, a backslash () can be preserved by setting this option to false.
SR_CONVERT_ISO7_FROM_UNIX=TRUE	If set, all textual body parts of messages coming in through the Internet Mail Gateway are converted to the ISO8859/1 character set, assuming the body parts contain IA5 characters with ISO-7 extensions. This option is only active if the SR_ISO7_HOST and SR_ISO7_*language* options are also present.
SR_CONVERT_ISO7_FROM_X400=TRUE	If set, all textual body parts of messages coming in through the X.400 Interface are converted to the ISO8859/1 character set, assuming the body parts contain IA5 characters with ISO-7 extensions. This option is only active if the SR_ISO7_HOST and SR_ISO7_*language* options are also present.
SR_CONVERT_ISO7_LANG=*language*	If set, activates the option SR_ISO7_*language* for messages passing through the Service Router. Only one instance of this option can be used and the language string must match a string in the ~/sys/LangMap file. See also UAL_ISO7_HOST.
SR_CONVERT_ONLY_IA5=TRUE	Used in conjunction with SR_CONVERT_ISO7_FROM_UNIX and SR_CONVERT_ISO7_FROM_X400. If set, only textual body parts with a character set of IA5 will be assumed to contain ISO-7 extensions and be eligible for conversion as specified by the options SR_ISO7_HOST and SR_ISO7_*language*.
SR_DUMP_MSGS=BEFORE or AFTER	Puts a copy of each message processed by the Service Router on the Dump Server queue DUMP. If the value of SR_DUMP_MSGS is set to BEFORE, the message is copied before it is processed by the Service Router. If the value of SR_DUMP_MSGS is set to AFTER, the message is copied after it is processed by the Service Router.

Option	Description
SR_EXPAND_PDL=TRUE	Sets the Service Router to perform Public Distribution List (PDL) expansion. When this option is set, the active distribution list of any message that is addressed to a PDL is expanded. (Expanded means a PDL entry is replaced by the full list of the recipients that it represents.) When a PDL has been expanded, the message is re-submitted to the Service Router. The Service Router will expand PDLs that *cannot* be routed regardless of whether this option is set or not. `SR_NO_ROUTE_PDL` stops expansion when a message cannot be routed.
SR_FILTER_TYPES_OF_ATT=TRUE	Causes the Service Router to remove `WINMAIL.DAT` attachments, used by some clients.
SR_ISO7_*language*= *ISO7_characters*	Specifies how text using the ISO-7 extensions is converted to the ISO8859/1 character set by the Service Router. *ISO7_characters* is a list of ISO8859/1 characters to which the 12 special "ISO-7" characters are mapped. The IA5 characters used as special ISO-7 characters are: #$@[]^‘{|}~ The ISO8859/1 equivalents (*ISO7_characters*) must be specified in the same order. Ensure that the ISO8859/1 equivalents are entered into the file using the ISO8859/1 character set! The *language* must correspond to the language set in the `SR_CONVERT_ISO7_LANG` option and the `SR_CONVERT_ISO7_LANG` option must be present to activate this option. See also `SR_CONVERT_ISO7_FROM_UNIX` and `SR_CONVERT_ISO7_FROM_X400`. Also `UAL_ISO7_`*language*.
SR_LD_BYPASS_LSERV=TRUE	When this option is set to `TRUE` (the default setting) the Service Router can bypass the Local Delivery Service and route a local message directly to the queue of one of the following Scalix services: Public Folder Server Directory Synchronization Server Error Manager Server Print Server Request Server By minimizing traffic through the Local Delivery Service, this option can reduce the amount of time required for Directory synchronization, and increase the speed of other local traffic. If an ACL is associated with the Local Delivery Service, you must set this option to `FALSE` to prevent the ACL being bypassed when a message is being routed directly to one of the Scalix services listed above.
SR_MAX_HOP_COUNT=*hop_count*	Specifies the number of hops that a message can make before it is assumed to be looping. The default is 100.

Option	Description
SR_MAX_NEST_LEVEL=*nest_level*	Specifies the maximum level of nesting allowed in a message before further nested parts are flattened by the Service Router. A value of zero means that all messages will be flattened. See also LD_MAX_NEST_LEVEL.
SR_NO_ROUTE_PDL=TRUE	Stops Public Distribution List (PDL) expansion by the Service Router when a message cannot be routed. (Normally, if a message cannot be routed when there is a PDL within the message's distribution list, the Service Router will expand the PDL, or PDLs, and try to route the message again before returning a Non-Delivery Notification.)
SR_Q_TIME_OUT=*seconds*	Specifies the time, in seconds, between checking the Service Router queue for new messages and checking for deferred messages that are due for submittal to the Service Router. The default is 30 seconds.
SR_RESOLVE_MASK=*number_of_ORname_fields*	Specifies the directory attributes that are retained in the recipient address when the address is automatically resubmitted by the Service Router following a delivery failure. These attributes are specified as internal or language dependant attribute tags separated by forward slashes (/). If a message cannot be routed or delivered using the full recipient address, you can resubmit the recipient names with a less fully specified address by specify how many O/R Name fields are retained when the name is resubmitted. For example: S/G/I/Q: This will retain only the Personal Names attributes (Surname, GivenName, Initials and Generation) CN/OU1: This will retain only the Common Name and OrgUnit 1 attributes

Option	Description
SR_ROUTE_X400_TO_OMX400_*n*= *route_match*	Allows messages to be rerouted from the `X400` queue to the `OMX400` queue for recipients whose address matches the specified values. The `X400` queue is used for messages routed to non-Scalix X.400 systems; the `OMX400` queue is used for messages routed to other Scalix systems. This option must be set on the Scalix system that contains the X.400 gateway where this rerouting is required. *n* is a number between 1 and 8. This enables you to specify up to 8 unique instances of this option in the General Configuration File. *route_match* specifies the route to be matched, using a series of O/R Address attributes and values, separated by forward slash characters (/). Attributes are specified as *TAG=value* pairings, where TAG is one of the following O/R Address attributes: TAG: O/R Address Attribute OU1: Organizational Unit Name 1 OU2: Organizational Unit Name 2 OU3: Organizational Unit Name 3 OU4: Organizational Unit Name 4 O: Organization Name P: Private Domain Name A: Administrative Domain Name C: Country OU1-TX: Teletex Organizational Unit Name 1 OU2-TX: Teletex Organizational Unit Name 2 OU3-TX: Teletex Organizational Unit Name 3 OU4-TX: Teletex Organizational Unit Name 4 O-TX: Teletex Organization Name The `value` specified is not case sensitive, and wildcard characters (*) can be used. If an attribute is not specified, it is treated as if it were fully wildcarded; that is, any value for that attribute is matched. No hierarchical rules are applied regarding which attributes can be specified and wildcarded.
SR_SYNC_P2_WITH_P1=TRUE	Sets the Service Router to modify the original value of the O/R Address in the P2 distribution list as well as the P1
SR_USEX500_DIR= TRUE or *X.500_Directory_Name*	Specifies that an X.500 Directory is used by the Service Router to resolve a DDN. If `SR_USEX500_DIR` is set to `TRUE`, the first X.500 Directory found is used. If the name of an X.500 Directory is specified, this Directory is used by the Service Router.

Option	Description
OMLIMIT_MIN_WARN_INTERVAL	NOTIFY messages for the OMLIMIT-EXCEEDED sanction are only sent if the NOTIFY message has not been sent within the time specified by this setting. The default value for the OMLIMIT_MIN_WARN_INTERVAL option is one day (1d). Example settings for this option are: 1h40m20s (1 hour 40 minutes and 20 seconds) 2d40 (2 days and 40 seconds) 6000 (6000 seconds/100 minutes) If the "omlimit -e u" sanction is enabled, the OMLIMIT_MIN_WARN_INTERVAL option also manages the interval during which omlimit-related messages are sent to a user.

UAL Client Interface Options

Option	Description
UAKD_CONNRATE_LIMIT= *number_of_connections_per_second*	Specifies the maximum number of client connection processes that can be started per second. Specify a value here in order to limit the rate at which client connections are attempted. This can prevent delays caused by connection processes waiting for Server resources.
UAKD_LISTEN_Q_SIZE= *number_of_connections*	Specifies the number of TCP/IP Socket connections that can be queued to the UAL Server listen.daemon process during busy periods. This reduces the possibility of "UAL unable to connect" errors. *number_of_connections* can be between zero and the operating system limit. The default is 20 connections.
UAKD_NICE_VALUE=20<=*value*<=20	Increases or reduces the priority of TCP/IP Socket connections to the UAL Server listen.daemon process, over other activities performed by the Scalix Server. *value* is a number between -20 and 20, where negative values increase the priority of client signon. The default is -10.
UAKD_SERVER_PUSH_NOTIFS= TRUE	Determines whether the Server-push mechanism is enabled. The default is TRUE. The Server-push mechanism allows certain clients to receive notifications automatically, without having to poll for them. Set this option to FALSE to disable the Server-push mechanism, forcing clients to poll for notifications. Setting this option to FALSE can result in increased performance, but do not set the option to FALSE if: There are a significant number of Outlook clients in use. This will cause a large increase in network traffic. Any IMAP clients are in use. IMAP clients cannot receive notifications if this option is FALSE.

Option	Description
UAL_5_40_PERF_CHANGES=TRUE	Switches on or off the performance changes to the UAL Client Interface that were introduced in Scalix Release 5.40. The default is TRUE. Set this option to FALSE if you suspect that one or more of the performance enhancements is causing problems.
UAL_ALLOW_DISABLED_CLIENTS=FALSE	If this is set to TRUE, those clients specified in the UAL_DISABLED_CLIENTS option are permitted to sign on to the Server. Such sign-ons are logged with a Warning logging level. This can be used to find out which users are using a particular client so that they can be warned before the client is actually disabled.
UAL_FLDR_ACL_DEFAULT = *permissions*	Specifies the permissions that are granted to the Default user when a Public Folder is created. These permissions will then apply to each user, unless the ACL has an entry that is more specific to that user. Set the value of *permissions* to a string of up to six characters, selected from the following: S (see), R (read), A (attach), D (delete), C (config). The default value for this option is S, R, A.
UAL_DEAD_TIMEOUT=*number_of_minutes*	Specifies the amount of time that Scalix will wait for the next UAL command of any type (as opposed to UAL_IDLE_TIMEOUT, which is triggered by active commands only) from a UAL client before assuming a timeout. Once the timeout period has been reached, Scalix assumes the connection to the client has been lost and signs off the user of the UAL client. If a timeout period is not specified, Scalix assumes the connection to the UAL client is good regardless of how long it has been waiting for another command. For serial connections, UAL_DEAD_TIMEOUT is overridden by UAL_SERIAL_DEAD_TIMEOUT. For local UAL clients, UAL_DEAD_TIMEOUT is overridden by UAL_LOCAL_DEAD_TIMEOUT.
UAL_DEF_MIME_MN_OVERRIDE= *route*	By default, a message generated using the Outlook client, with the "Send in RTF" flag unchecked, will be routed according to the default MIME mailnode entry in the Routing Table. Set this option if you want to use a different route for such messages. The route you specify must ultimately point to a MIME gateway. Specify a route as an OR address pattern, in the format specified for the -m option in the omaddrt man page. For example, UAL_DEF_MIME_MN_OVERRIDE="internet,ux". If you specify the address pattern in this route as a Teletex value, you must specify an appropriate display character set in the UAL_DEF_MIME_MN_OVERRIDE_CS option.
UAL_DEF_MIME_MN_OVERRIDE_CS=*character-set*	This option specifies the display character set for the address pattern you specify in the UAL_DEF_MIME_MN_OVERRIDE option, if you entered it as a Teletex value. The default character set is ISO8859_1. This option has no effect if the UAL_DEF_MIME_MN_OVERRIDE option is not set.

Option	Description
UAL_DEF_TNEF_MN_OVERRIDE=*route*	By default, a message generated using the Outlook client, with the "Send in RTF" flag checked, will be routed according to the default TNEF mailnode entry in the Routing Table. Set this option if you want to use a different route for such messages. The route you specify must ultimately point to a TNEF gateway. Specify a route as an OR address pattern, in the format specified for the `-m` option in the `omaddrt` man page. For example, `UAL_DEF_TNEF_MN_OVERRIDE="internet,ux"`. If you specify the address pattern in this route as a Teletex value, you must specify an appropriate display character set in the `UAL_DEF_TNEF_MN_OVERRIDE_CS` option.
UAL_DEF_TNEF_MN_OVERRIDE_CS=*character-set*	This option specifies the display character set for the address pattern you specify in the `UAL_DEF_TNEF_MN_OVERRIDE` option, if you entered it as a Teletex value. The default character set is ISO8859_1. This option has no effect if the `UAL_DEF_TNEF_MN_OVERRIDE` option is not set.
UAL_DIR_LIST_SORT_ORDER=*list_of_internal_attributes*	Specifies the order in which Directory attributes are sorted. The order is specified as a list of internal attribute names with each attribute separated by a /. The internal attribute names, which are numbers for the core Scalix attributes, are listed using the command `omshowatt -u`.
UAL_DIR_LIST_SORT_PROG=*absolute_program_name*	Specifies the program that sorts lists of Directory entries for UAL clients. The value *absolute_program_name* must specify the full path name of the sorting program together with any parameters that are necessary. The default Scalix sort program is `/bin/sort -f`. This is used if `UAL_DIR_LIST_SORT_PROG` is not set.
UAL_DIR_MOD_FULL_NAME=TRUE	Specifies that Full Name Checking is always done on the `UAL_CHKLST`, `UAL_CHKNAM`, `UAL_DELENT` and `UAL_MODENT` commands.
UAL_DISABLE_BB=FALSE	Disables or enables public folder access. If this tag is set to `TRUE`, then if the user attempts to perform an action involving Public Folders the client displays an error message stating that the user has insufficient access capabilities to perform the action. The default is `FALSE`.
UAL_DISABLED_CLIENTS=*strings*	Specifies those UAL clients that are disabled from signing on to the Server. *strings* is a list of space-separated, quoted strings (use single quotes only). Each string is a client identity string as passed in the `UAL_INIT` command. Strings can contain wildcards. Sign-on attempts by identified clients are refused, and logged with an Error logging level. See also `UAL_ALLOW_DISABLED_CLIENTS`.
UAL_DISABLE_NESTED_BBS=TRUE	Stops the UAL Client Interface creating new nested Public Folders under top-level Public Folders.

Option	Description
UAL_DISALLOW_AUTO_PASSWORD=TRUE	If set, a client cannot sign on to Scalix if the client has explicitly indicated that its password was obtained from a configuration file rather than having been entered interactively by a user. See also UAL_DISALLOW_NON_USER_PASSWORD. The user-specific setting of this option overrides the system-wide setting. Note that this mechanism is not intended to provide a secure indication.
UAL_DISALLOW_NON_USER_PASSWORD=TRUE	If set, a client cannot sign on to Scalix if the client has *not* explicitly indicated that its password was obtained interactively from a user. See also UAL_DISALLOW_AUTO_PASSWORD. Note that this option will only work with clients that supply the "password origination status". If a client does not support this element, then it will not be able to sign on even if the password is actually entered interactively by the user. The user-specific setting of this option overrides the system-wide setting. Note that this mechanism is not intended to provide a secure indication.
UAL_DL_SIZE_LIMIT=*no_of_kilobytes*	Sets the Distribution List area size limit. in kilobytes. A value of zero (0) means no size limit. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_DO_LONG_INET_CHECK=FALSE	Specifies whether the POP3 and IMAP4 Servers perform full checking of Internet addresses. The default value is FALSE, and this causes the POP3 and IMAP4 Servers to only look to see if a name has a DDA of type RFC-822 when checking if there is an Internet version of the name. This allows greater efficiency in cases where the names are either in a DDA or held in a Directory, since a check to see if the name is routable to a Unix queue is omitted. Set this option to TRUE to cause the full range of address conversions to be applied (according to the unixmap.in and unixin.rules steering files).
UAL_FC_SIZE_LIMIT=*no_of_kilobytes*	Sets the Filing Cabinet size limit. The value is set in kilobytes. A value of zero means no size limit. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.

Option	Description
UAL_FORCE_IA_IN_ORN=FALSE	Determines whether the UAL is forced to put the Internet address, as configured in the SYSTEM Directory, of all DL names in the message, doing additional directory lookups if necessary. Assuming that the SYSTEM Directory is populated with Internet addresses, when this option is set to FALSE (the default), the UAL will insert the Internet addresses of DL entries in the message, where it can do so without additional directory lookups. When a message reaches the Internet Mail Gateway, or is browsed by a POP3 or IMAP4 client, any ORNs without an Internet address are looked up in the Directory to retrieve the Internet address. So the Directory lookup overhead is at the Internet Mail Gateway, the POP3 or the IMAP4 interfaces. Set this option to TRUE to switch the overhead to the time when the message is sent.
UAL_FORCE_TRACE_LEVEL=*trace_level*	Sets the UAL trace level on a system-wide basis, overriding any trace value supplied by a client or set in the `user.cfg` file. *trace_level* can be any valid trace level, including 0 (zero), which switches off tracing.
UAL_GIVE_GROUP5_INET_NAME_STRICT=FALSE	Determines whether the UAL Server gives a Group 5 Internet name in all cases. When set to FALSE (the default), the UAL Server gives a Group 5 Internet name in all cases where the address contains a DDA of type RFC-822. This occurs even for those addresses that are tunnelled through Scalix, and causes replies to be incorrectly routed through the default Internet Mail Gateway. Set the option to TRUE to cause the UAL Server to only give a Group 5 Internet name when the address in the DDA is routable to the local Internet Mail Gateway. There are two circumstances under which setting this option to TRUE will have no effect: If Scalix users and PDLs have Internet addresses configured. If Internet addresses of external users are put into the Group 5 Internet address at the incoming Internet Mail Gateway. You can prevent this by setting the option `INET_NO_IA_IN_ORN` to TRUE.
UAL_IDLE_SHUTDELAY=*number_of_minutes*	Specifies the additional delay in shutting down a UAL client connection that has timed out. `UAL_IDLE_SHUTDELAY` is used with `UAL_IDLE_TIMEOUT`. See also `UAL_IDLE_TIMEOUT`. For serial connections, `UAL_IDLE_SHUTDELAY` is overridden by `UAL_SERIAL_IDLE_SHUTDELAY`. For local UAL clients, `UAL_IDLE_SHUTDELAY` is overridden by `UAL_LOCAL_IDLE_SHUTDELAY`.

Option	Description
UAL_IDLE_TIMEOUT=*number_of_minutes*	Specifies the amount of time that Scalix will wait for the next *active* UAL command from a UAL client before assuming a timeout (`PREPARE MESSAGE`, `ATTACH ITEM` are examples of active UAL commands, and `NEW MESSAGES` and `LIST ACK` are examples of passive UAL commands). Once the timeout period has been reached, Scalix assumes the connection to the client has been lost and signs off the user of the UAL client. (The sign off can be delayed using `UAL_IDLE_SHUTDELAY`.) If a timeout period is not specified, Scalix assumes the connection to the UAL client is good regardless of how long it has been waiting for another command. `UAL_IDLE_TIMEOUT` is used with `UAL_IDLE_SHUTDELAY`. For example: If `UAL_IDLE_TIMEOUT` is set to 30 minutes, and `UAL_IDLE_SHUTDELAY` is not set, the client is disconnected from the Server 30 minutes after the last active UAL command was issued. If `UAL_IDLE_TIMEOUT` is set to 30 minutes, and `UAL_IDLE_SHUTDELAY` is set to 10 minutes, 30 minutes after the last active UAL command is issued, the client displays a dialog box asking if the user wants to retain the connection. This dialog box is displayed for up to the 10 minutes specified by `UAL_IDLE_SHUTDELAY`. If the user responds within this time with a Yes, that is considered an active UAL command, and the `TIMEOUT` countdown restarts from the beginning. If the user responds with a No, the connection is closed. If the user does not respond within the 10 minutes, the connection is closed. For serial connections, `UAL_IDLE_TIMEOUT` is overridden by `UAL_SERIAL_IDLE_TIMEOUT`. For local UAL clients, `UAL_IDLE_TIMEOUT` is overridden by `UAL_LOCAL_IDLE_TIMEOUT`.
UAL_INTRAY_SIZE_LIMIT=*no_of_kilobytes*	Sets the In Tray size limit in kilobytes. A value of zero (0) means no size limit. Set Message Store size limits using the `omlimit` command. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_ISO7_FROM_HOST=*language*	This option is the same as `UAL_IS07_HOST` except that the character set conversion only occurs when text is passed from Scalix to the client and not when it is passed back to the Server.

Option	Description
UAL_ISO7_HOST=*language*	This option allows clients to interoperate with a Scalix Message Store containing IA5 text that uses the ISO-7 extensions. If set, activates the option UAL_ISO7_*language* for any clients using *language*. IA5 text with the ISO-7 extensions are converted to the ISO8859/1 character set when downloaded to or displayed by a client and conversely, the ISO8859/1 characters are mapped back into IA5 with ISO-7 extensions when entering the Scalix system from a client. Only one instance of this option can be used and the *language* string must match a string in the ~/sys/LangMap file. See also UAL_ISO7_FROM_HOST, UAL_ISO7_TO_HOST, and SR_ISO7_HOST.
UAL_ISO7_*language*= *ISO7_characters*	Specifies how text using the ISO-7 extensions is converted to the ISO8859/1 character set. *ISO7_characters* is a list of ISO8859/1 characters to which the 12 special "ISO-7" characters are mapped. The IA5 characters used as special ISO-7 characters are: #$@[\]^`{|}~ The ISO8859/1 equivalents (*ISO7_characters*) must be specified in the same order. Ensure that the ISO8859/1 equivalents are entered into the file using the ISO8859/1 character set! The *language* must correspond to the language set in the UAL_ISO7_HOST option and the UAL_ISO7_HOST option must be present to activate this option. See also SR_ISO7_*language*. UAL_ISO7_TO_HOST=*language* This option is the same as UAL_ISO7_HOST except that the character set conversion only occurs when text is passed from the client to Scalix and not when it is passed back to the client.
UAL_KILL_REMOTE_SIGNON_2= TRUE	Allows the Scalix Server to kill a current user session in order to allow the user to sign on again. If a user session is terminated abnormally (for example, if a user reboots their PC), the session can continue to exist on the Server. This could prevent the user from signing on again. Setting this option to TRUE allows the Scalix Server to kill the user's oldest session, so allowing the user to sign on. The Scalix Server permits 17 concurrent signons. If this option is set to TRUE, and the user tries to connect for the eighteenth time, the Scalix Server kills the user's oldest session, and then allows them to sign on again. If the option is set to FALSE, the Scalix Server will not permit the user to sign on. Note that some clients can set a lower limit for the number of concurrent signons.

Option	Description
UAL_LIST_CACHE_SIZE= *number_of_message_parts*	Specifies the number of message parts that can be held in memory by a UAL process. This entry reduces I/O by forcing Scalix to keep the message in memory instead of creating and then opening one file for each message part and the message header. The default is 4, which equates to a header record and three body parts.
UAL_LOCAL_DEAD_TIMEOUT= *number_of_minutes*	Specifies the amount of time that Scalix will wait for the next UAL command of any type (as opposed to UAL_LOCAL_IDLE_TIMEOUT, which is triggered by active commands only) from a *local* UAL client before assuming a timeout. Once the timeout period has been reached, Scalix assumes the connection to the client has been lost and signs off the user of the UAL client. If a timeout period is not specified, Scalix assumes the connection to the local UAL client is good regardless of how long it has been waiting for another command. UAL_LOCAL_DEAD_TIMEOUT overrides UAL_DEAD_TIMEOUT. To remove a timeout for local UAL clients that was set using UAL_DEAD_TIMEOUT, set UAL_LOCAL_DEAD_TIMEOUT to 0.
UAL_LOCAL_IDLE_SHUTDELAY= *number_of_minutes*	Specifies the additional delay in shutting down a local UAL client connection that has timed out. UAL_LOCAL_IDLE_SHUTDELAY is used with UAL_LOCAL_IDLE_TIMEOUT.
UAL_LOCAL_IDLE_TIMEOUT= *number_of_minutes*	Specifies the amount of time that Scalix will wait for the next *active* UAL command from a local UAL client before assuming a timeout (PREPARE MESSAGE, ATTACH ITEM are examples of active UAL commands, and NEW MESSAGES and LIST ACL are examples of passive UAL commands). Once the timeout period has been reached, Scalix assumes the serial connection to the client has been lost and signs off the user of the UAL client. (The sign off can be delayed using UAL_LOCAL_IDLE_SHUTDELAY.) If a timeout period is not specified, Scalix assumes the connection to the UAL client is good regardless of how long it has been waiting for another command. UAL_LOCAL_IDLE_TIMEOUT is used with UAL_LOCAL_IDLE_SHUTDELAY. UAL_LOCAL_IDLE_TIMEOUT overrides UAL_IDLE_TIMEOUT. To remove a timeout for local UAL clients that was set using UAL_IDLE_TIMEOUT, set UAL_LOCAL_IDLE_TIMEOUT to 0.
UAL_LOCAL_IGNORE_PASSWORD= TRUE	Removes the password entry stage from the sign on process. The sign on will succeed only if the user has logged in using their Scalix mailbox Linux login, and if the user is using a local UAL Client.

Option	Description
UAL_MAP_ALIAS_AT_MAIL=FALSE	By default, recipient O/R Addresses entered as aliases in a distribution list are displayed as those aliases to the recipients. Set this option to TRUE to cause aliases to be rewritten as their *real* O/R Addresses when the message is submitted to Scalix (unless both sender and recipient are using the Outlook mail client). This enables the user to use and see alias names when preparing a message, but the recipients of the message only see the *real* names in the distribution list, not the alias names. If both sender and recipient are using the Outlook mail client, then setting this option to TRUE has no effect, and the recipients continue to see the alias names in the distribution list.
UAL_MAX_SIGNON_PER_USER=*number*	Specifies the number of simultaneous signons that a user can have. The default is 17.
UAL_MOD_BB_ITEMS=TRUE	Determines whether items attached to Public Folders can be modified. Set this option to FALSE to prevent modification of Public Folder items. In this case, users can still be able to add top-level items to Public Folders, or delete top-level items from Public Folders, depending on the Public Folder's ACL. When set to TRUE (the default), master items can be modified, although slave items cannot. Public folders can be accessed only by Premium users. For more information, see "About Scalix Product Editions".
UAL_MSTORE_SIZE_LIMIT=*no_of_kilobytes*	Sets the overall message store size limit. The value is set in kilobytes. A value of zero means no size limit. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_NAMED_PIPE_BLOCK_SIZE=*block_size*	Sets the physical block size for TCP/IP Named Pipes client connections. The default is 1380 bytes.
UAL_NMP_DELAY=*number_of_milliseconds*	Specifies the time delay before returning a reply to a command from a UAL client using a TCP/IP Named Pipes connection. By default, there is no time delay, but this can mean the receiving client system can *miss* the reply when it is sent. To overcome this problem set the delay to between 10 and 100 milliseconds.
UAL_NO_AUTOGEN_IA=FALSE	Determines, for certain UAL clients, whether automatic Internet address mapping is enabled. This option only affects those UAL clients that use the UAL_ENTADD command to add Directory entries. Set this option to TRUE to override automatic Internet address mapping for these UAL clients. The default is FALSE. See the option INET_USE_AUTO_IAM.
UAL_NO_DESIGNATE_SIGNON=TRUE	Removes the designate sign on feature.

Option	Description
UAL_NO_IA_IN_ORN=FALSE	Determines whether the UAL puts the Internet addresses of the sender, recipients or DL names, into the message. When set to FALSE (the default), Internet addresses are inserted into the message if they can be determined without additional directory lookups.
UAL_NO_REPLY_BLOCKING=TRUE	If set, multiple UAL Client Interface replies are not blocked up before being sent to a UAL remote client. This is used to overcome data-comm problems that result from large blocks being sent from the Server.
UAL_NO_WB_EMPTY=TRUE	Stops a user's Waste Basket being emptied when the user has finished using a UAL client and signs off. If this option is set, use the command `omtidyu` or `omtidyallu` to ensure Waste Baskets continue to be emptied regularly.
UAL_OUTTRAY_SIZE_LIMIT=*no_of_kilobytes*	Sets the Out Tray size limit in kilobytes. A value of zero (0) means no size limit. Set Message Store size limits using the `omlimit` command. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_PASSWORD_AGED=IGNORE WARN or ERROR	This option determines the effect of an expired password on a user signing on to Scalix through a client. The default value is ERROR. If the user's password has expired, an error is generated when the user attempts to signon and the signon fails. The signon can only succeed when a valid new password is supplied. If the value is set to WARN and the user's password has expired, the user can sign on using the expired password but a warning message is placed in their In Tray stating that their password has expired and should be changed immediately. (This message appears in the In Tray for the first signon of the day.) If the value is set to IGNORE any user password expiration condition is ignored (a Scalix user will be allowed to signon even if their password has expired.)
UAL_PEND_SIZE_LIMIT=*no_of_kilobytes*	Sets the Pending Tray size limit in kilobytes. A value of zero (0) means no size limit. Set Message Store size limits using the `omlimit` command. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_POP3_HOSTNAME=*hostname*	Set the POP3 Server or the IMAP4 Server hostname so that addresses display as `name@hostname`. By default, the name of the Scalix host is used.
UAL_POP3_LANG=*language*	Specifies the Scalix language to use for error messages returned by the POP3 or IMAP4 client. The default is C.

Option	Description
UAL_POP3_TIMEOUT=*no_of_seconds*	Specifies the number of seconds of inactivity allowed before connection to the POP3 Server will timeout. The default is 600 seconds.
UAL_POP3_TRACE=TRUE	If set, information from the `in.pop3d` process is traced and placed in the `~Scalix/tmp` directory. You can set this option to `DETAIL` to generate more detailed logging. Set the option to `FALSE` to prevent logging.
UAL_PRINT_SERVER_ONLY=TRUE	If set, all printing by UAL clients goes through the Print Server. See also `UAL_PRINT_SPECIFICATION`.
UAL_PRINT_SPECIFICATION=*print_command*	If set, *print_command* overrides any printer specification supplied by a UAL client. The *print_command* can either be a Linux printer command line or a Print Server printer specification.
UAL_PWD_WARNING_DAYS=*days*	Activates the mechanism to generate advisory messages to users whose mailbox passwords are due to expire within the period specified by `days`. The warning message appears as a new message in the user's In Tray for the first signon of the day. Use this option if clients are being used that do not recognize the *password expired* signon error. These clients cannot signon successfully once the user's password has expired.
UAL_SCK_DELAY=*number_of_milliseconds*	Specifies the time delay before returning a reply to a command from a UAL client using a TCP/IP Sockets connection. By default, there is no time delay, but this can mean the receiving client system can *miss* the reply when it is sent. To overcome this problem set the delay to between `10` and `100` milliseconds.
UAL_SEND_OBJECT_FILES=TRUE	Determines whether an object file (created with the UAL `addobj` call) is mailed with the message to which it is attached. When set to `TRUE`, the UAL submits both the message and any attached object files (assuming that the object files do not have the `UAL_ADDOBJ_NOT_MAIL` flag set).
UAL_SERIAL_DEAD_TIMEOUT=*number_of_minutes*	Specifies the amount of time that Scalix will wait for the next UAL command of any type (as opposed to `UAL_SERIAL_IDLE_TIMEOUT`, which is triggered by active commands only) from a UAL client *using a serial connection* before assuming a timeout. Once the timeout period has been reached, Scalix assumes the connection to the client has been lost and signs off the user of the UAL client. If a timeout period is not specified, Scalix assumes the *serial* connection to the UAL client is good regardless of how long it has been waiting for another command. `UAL_SERIAL_DEAD_TIMEOUT` overrides `UAL_DEAD_TIMEOUT`. To remove a timeout for UAL clients that was set using `UAL_DEAD_TIMEOUT`, set `UAL_SERIAL_DEAD_TIMEOUT` to `0`. This removes the timeout for all UAL clients using a serial connection.

Option	Description
UAL_SERIAL_IDLE_SHUTDELAY= *number_of_minutes*	Specifies the additional delay in shutting down a UAL client *serial* connection that has timed out. UAL_SERIAL_IDLE_SHUTDELAY is used with UAL_SERIAL_IDLE_TIMEOUT.
UAL_SERIAL_IDLE_TIMEOUT= *number_of_minutes*	Specifies the amount of time that Scalix will wait for the next *active* UAL command from a UAL client *using a serial connection* before assuming a timeout (PREPARE MESSAGE, ATTACH ITEM are examples of active UAL commands, and NEW MESSAGES and LIST ACL are examples of passive UAL commands). Once the timeout period has been reached, Scalix assumes the serial connection to the client has been lost and signs off the user of the UAL client. (The sign off can be delayed using UAL_SERIAL_IDLE_SHUTDELAY.) If a timeout period is not specified, Scalix assumes the connection to the UAL client is good regardless of how long it has been waiting for another command. UAL_SERIAL_IDLE_TIMEOUT is used with UAL_SERIAL_IDLE_SHUTDELAY. UAL_SERIAL_IDLE_TIMEOUT overrides UAL_IDLE_TIMEOUT. To remove a timeout for UAL clients that was set using UAL_IDLE_TIMEOUT, set UAL_SERIAL_IDLE_TIMEOUT to 0. This removes the timeout for all UAL clients using a serial connection.
UAL_SET_ROC_ON_ND= TRUE or FALSE	Overrides client setting of "Return of contents on non-delivery". Set to TRUE to request a return of contents for all non-delivered messages. Set to FALSE to prevent return of contents for all non-delivered messages.
UAL_SHOW_8BIT_T61_AS_1167= TRUE	If set, teletex body parts (file code 1736) are presented to clients as normal 8-bit text (file code 1167). This enables existing Western European clients to work with no loss of features when handling teletex body parts that contain 8-bit character sets.
UAL_SIGNON_ALIAS=YES or ONLY	Specifies whether aliases are used for sign on. Any UAL_SIGNON_ALIAS entries in user.cfg take precedence over the UAL_SIGNON_ALIAS entry in general.cfg. (This enables you to set a default use of aliases in general.cfg and then set overrides for specific users in user.cfg.) The YES value means aliases can be used to sign on with, users can also continue to use their Personal Name if they want to. The ONLY value means the aliases only can be used to sign on with, the Personal Name cannot be used any more. UAL_SIGNON_ALIAS is used with UAL_SIGNON_ALIAS_CONFIG and UAL_USE_SIGNON_ALIAS.

Option	Description
UAL_SIGNON_ALIAS_CONFIG= SYS or USER	Allows users to sign on using an alias. The SYS value means that everyone can sign on using an alias. The USER value means that alias sign on entries in `user.cfg` will be used when they exist, and will take precedence over any alias sign on entries in `general.cfg`. `UAL_SIGNON_ALIAS_CONFIG` is used with `UAL_SIGNON_ALIAS` and `UAL_USE_SIGNON_ALIAS`.
UAL_SINGLE_TEMP_DIR= *temp-directory*	Specifies the location of user temporary files. Normally (when this option is not set), the user temporary files are stored in the directory files for each user. Specify a value for this option to cause all user temporary files to be written to a single directory. This allows you to use a high-speed (and possibly low-recovery) file system (for example, a RAM disk) to store these temporary files. The directory you specify must have: permissions 771 an owner of `scalix` a group of `scalix` a path length of 225 characters or less For example, to create a directory called `usr_tmp`, enter the following commands: mkdir $(omrealpath '~/usr_tmp') chown scalix:scakix $(omrealpath '~/usr_tmp') chmod 771 $(omrealpath '~/usr_tmp')
UAL_SIZE_ERR_TO_USER=TRUE	Specifies that a UAL error message is generated when a user tries to create an item in their Filing Cabinet or Distribution List area once it has exceeded the limit set by `omlimit`.
UAL_SIZE_MSG_TO_ENU=TRUE	Specifies that a message is sent to the Error Notification user when a user's In Tray, Pending Tray or Waste Basket exceeds the set warning limit, boundary limit, or maximum limit. See `UAL_SIZE_WARNING_BOUNDS` and `UAL_SIZE_WARNING_LIMIT`.
UAL_SIZE_MSG_TO_USER=TRUE	Specifies that a message is sent to the user when their In Tray, Pending Tray or Waste Basket exceeds the set warning limit, boundary limit, or maximum limit. See `UAL_SIZE_WARNING_BOUNDS` and `UAL_SIZE_WARNING_LIMIT`.
UAL_SIZE_ON_RECEIPT=FALSE	Specifies whether a user whose message store components exceed their configured limits is prevented from receiving messages. When this option is set to FALSE (the default), users are not prevented from receiving messages even if the size of their message store component is greater than its configured limit.

Option	Description
UAL_SIZE_ON_SEND=FALSE	Specifies whether a user whose message store components exceed their configured limits is prevented from sending messages. When this option is set to `TRUE`, then message delivery rules can be implemented that limit a user's ability to send messages. These rules utilize the `OMLIMIT-EXCEEDED` message attribute filter. When this option is set to `FALSE` (the default), then rules based on the `OMLIMIT-EXCEEDED` filter have no effect.
UAL_SIZE_WARNING_BOUNDS=*percent_increase*	Specifies the boundary levels for warnings between the warning limit and the maximum limit. For example, if set to `5`, warnings will be sent when the size of an In Tray, Pending Tray or Waste Basket increases by 5% or more since the last warning. To enable warning messages to be sent, you must have set at least one of these options: `UAL_SIZE_MSG_TO_ENU` and `UAL_SIZE_MSG_TO_USER`.
UAL_SIZE_WARNING_LIMIT=*percentage_of_max_limit*	Specifies the percentage of the maximum limit, set on the size of the In Tray, Pending Tray or Waste Basket, that should be reached before a warning messages is generated. For example, if set to `80`, warnings will be sent when the In Tray, Pending Tray or Waste Basket area reaches 80 percent of its maximum limit. To enable warning messages to be sent, you must have set at least one of these options: `UAL_SIZE_MSG_TO_ENU` and `UAL_SIZE_MSG_TO_USER`.
UAL_SOCKET_BLOCK_SIZE=*block_size*	Sets the physical block size for Sockets client connections. The default is 1380 bytes.
UAL_TTX_NAME_FORMAT_LANG=*attribute order*	Specifies the order in which Personal Name attributes are displayed for clients using the UAL Client Interface display program (`item.browse`). The four Personal Name Attributes are represented with the following letters: `S`: Surname `F`: Given Name `I`: nitials `G`: Generation Qualifier Enter the letters in the order you want the attributes to be displayed. The *LANG* part of the option specifies the language the format is used for. For example, to display native Japanese names in their natural form, specify the option like this: UAL_TTX_NAME_FORMAT_NIPPON=SF
UAL_TTX_NAME_SHOW_ALL=TRUE	Sets the UAL Client Interface display program (`item.browse`) to display all teletex O/R Address attributes regardless of whether the correct client character set is configured. By default, these address attributes are not displayed unless a suitable client character set is configured.

Option	Description
UAL_TTX_NAME_SUBST=TRUE	Substitutes the teletex O/R Address attributes with the corresponding printable string attributes before displaying the message. This option is for clients using the UAL Client Interface display program (`item.browse`).
UAL_USE_SIGNON_ALIAS=FALSE or TRUE	Specifies whether the alias is used after sign on. If you set `UAL_USE_SIGNON_ALIAS` to `FALSE`, the UAL client reverts to using the user's Personal Name for the remaining time the user is signed on (the alias or Personal Name is used on the "Creator" part of a message). If you set `UAL_USE_SIGNON_ALIAS` to `TRUE`, the alias is used for the remaining time the user is signed on. `UAL_USE_SIGNON_ALIAS` is used with `UAL_SIGNON_ALIAS` and `UAL_SIGNON_ALIAS_CONFIG`.
UAL_WB_SIZE_LIMIT=*no_of_kilobytes*	Sets the Waste Basket size limit in kilobytes. A value of zero (0) means no size limit. Set Message Store size limits using the `omlimit` command. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect. UXO_CHECK_TYPES_OF_DDA=*DDA_type* Specifies the DDA types that are acceptable as valid internet addresses, to enable the UAL Client Interface and the Internet Mail Gateway to route the message to the correct destination for the recipient. *DDA_type* is one of the following: One or more valid DDA types, for example: RFC-822 HPMEXT1 HPMEXT2 HPMEXT3 HPMEXT4 You can specify up to 10 DDA types; if you do specify more than one type, separate them with commas. For example: RFC-822,HPMEXT1,HPMEXT2,HPMEXT3,HPMEXT4 FALSE No DDA type checking is performed. This behavior is as for Scalix systems before B.05.10. If your Scalix Directory contains DDAs with no type specified and they are not valid internet addresses, you are recommended to set this option to: UXO_CHECK_TYPES_OF_DDA=RFC-822,HPMEXT1, HPMEXT2,HPMEXT3,HPMEXT4 If this option is not present, the default setting is: UXO_CHECK_TYPES_OF_DDA=,RFC-822,HPMEXT1, HPMEXT2,HPMEXT3,HPMEXT4 The leading comma means that DDAs with no type specified are also acceptable as valid internet addresses.

Option	Description
OMLIMIT_MIN_WARN_INTERVAL	If the "`omlimit -e u`" sanction is enabled, the `OMLIMIT_MIN_WARN_INTERVAL` option also manages the interval during which `omlimit`-related messages are sent to a user. The default value for the `OMLIMIT_MIN_WARN_INTERVAL` option is one day (`1d`). Example settings for this option are: `1h40m20s` (1 hour 40 minutes and 20 seconds) `2d40` (2 days and 40 seconds) `6000` (6000 seconds/100 minutes) The `OMLIMIT_MIN_WARN_INTERVAL` option also manages the interval between `OMLIMIT-EXCEEDED` notifications when the service router processes message delivery rulesets.

Virus Protection Options

Option	Description
SR_VS_DO_VIRUS_SCAN=FALSE	In the absence of the `ALL-ROUTES.VIR` ruleset file, this option determines whether virus scanning is active. If the `ALL-ROUTES.VIR` file exists, then the rules within that file determine the virus scanning/cleaning action that will be taken. Set the option to `TRUE` to cause the Service Router to check all message attachments for viruses. If a virus is found, the message is not routed, and Scalix generates a non-delivery notification. Set the option to `FALSE` to disable virus checking. Note that the performance of your Scalix system can be degraded if you enable virus checking and a large number of viruses are detected, since each virus detected will cause Scalix to generate a non-delivery report.
SR_VS_IGNORE_ITEM_TYPES=*filetype-no*	Specifies the filetypes of items that will not be scanned for viruses. By default, when virus scanning is enabled, either by setting the `SR_VS_DO_VIRUS_SCAN` option to `TRUE` or by creating the `ALL-ROUTES.VIR` ruleset file, all filetypes are scanned for viruses. Use this option to prevent certain filetypes from being scanned. *filetype-no ...* is a colon-separated list of numerical file codes, as specified in `~/nls/`*language*`/filetype.` For example, set `SR_VS_IGNORE_ITEM_TYPES` to `1167` to prevent text files from being scanned.
SR_VS_TEST_SCAN_SL=*string*	Specifies the location of the test virus shared library. If this file is in its default location, you must set this option to `/opt/scalix/`*version*`/lib/libom_testvs.sl` if you want to test your virus scanning configuration.

Option	Description
SR_VIRUS_SCAN_TYPE=*string*	Specifies whether virus checking is operating in test mode. Set this option to "`Test Scan`" to cause the Service Router to check messages and generate a non-delivery notification if the first five characters of any attachment is `VIRUS`. Note: if you set this option to `"Test Scan"`, you must also set the `SR_VS_TEST_SCAN_SL` option to the location of the test virus shared library. If you set this option to "`Generic`", `~/sys/omvscan.cfg` determines the virus scanning engine to use. Also, you must copy `omvscan.map` to the `~/rules/` directory to enable virus scanning.

Export Process Options

Option	Description
XP_START_IMPORT_DELAY=*seconds*	If set, specifies the number of seconds that the `xport.in` process will wait when invoked by the Service Router in recovery mode before starting to process messages and put them on the Service Router queue. The delay is only observed to a resolution of 5 seconds. A value of zero will cause `xport.in` to start processing the messages immediately. The default is 60 seconds.

Miscellaneous Options

Option	Description
AK_ACK_MSG_PRI=2	Determines the priority to use for acknowledgments if `AK_ACK_SAME_PRI` is not set or the priority of the message being acknowledged is not known. Valid values are 0 (normal), 1 (non-urgent), and 2 (urgent). The default is 2 (urgent).
AK_ACK_SAME_PRI=TRUE	When set to `TRUE`, this option causes the priority of an acknowledgment to be the same as that of the message being acknowledged where that can be determined. Otherwise, the value of `AK_ACK_MSG_PRI` is used. The default is `FALSE`.

Option	Description
CT_OLD_PENDING_SIZE_MODE= FALSE	Determines whether items in the message store that are pending deletion are included in the reported size of the message store. When this option is set to TRUE, then items that are pending deletion are included in the reported size. The size is not reduced until the items are actually deleted. Leave this option FALSE (the default) if you want the reported size of the message store to be reduced as soon as items are marked for deletion. This is useful, for example, when users have limits configured on their message store sizes: if set to FALSE, users will see an immediate effect on their reported message store sizes when they delete items. If set to TRUE, the message store size will not change until the user logs out of Scalix (for multiple signons, the last user must log out). If you change the value of this option, you should run the `omscan -A -S` command to regularize the reported message store sizes.
CT_OLD_SIZE_METHOD=FALSE	Specifies how the size of containers in the message store is updated. When set to FALSE (the default), container sizes are written to the Container Access Monitor, so that all processes have access to them. This ensures that all processes use the same reported container sizes, and is particularly useful when message store size limits are configured. If set to TRUE, container sizes are stored within an individual process before being written to disk.
DIA_NO_T61_SUBJECT=FALSE	Determines whether the `omcontain` command attempts to display T61 subjects. When set to FALSE (the default), omcontain does not display T61 subjects.
IM_MAKE_MSG_ID_GLOBAL_ UNIQUE=TRUE	Specifies that Scalix message IDs should use the long format. Set the value to FALSE if you want message IDs to use the short format. Note, however, that this can cause message IDs not to be globally unique. If you change the value of this option, you must restart Scalix for the change to take effect.
SMTPD_PWD_TRANSITION=FALSE	Determines whether the SMTP relay generates SASL passwords and stores them in the user list. Set this option to TRUE to cause the SMTP Relay to generate alternative SASL passwords when a PLAIN password authentication succeeds. These SASL passwords are then stored in the user list. Reset this option to FALSE (the default) when the user list contains all the SASL passwords.
USRL_AUTO_GEN_AUTHID=G_I_S	Specifies the method used to generate the name part of the Authentication ID. If you want to use the OMID method instead, enter DEFAULT for this value.

Client-specific Configuration Options

A subset of the options used in the general.cfg file can be specified for individual Scalix clients. The options for each client are held in a file with the name of the client host's Fully Qualified Domain Name (FQDN), in the directory ~/sys/client.cfg. For example, for a client on the host north.sales.alpha.com, the client configuration options file would be named ~/sys/client.cfg/north.sales.alpha.com.

Note	Scalix Connect for Microsoft Outlook can be used only by Premium users. For more information, see "About Scalix Product Editions".

The client.cfg directory does not exist by default and must be created. It should be owned by the user scalix with permissions of 555 (dr-xr-xr-x). Client files within this directory should be owned by the user scalix with permissions of 444 (-r--r--r--).

See the descriptions in the <Xref_Color>"System-wide Configuration Options" section for the following options:

- IMAP_AUTOMATIC_MDN=FALSE
- IMAP_BB_FOLDER_PREFIX=#bb
- IMAP_BB_FOLDER_SEPARATOR=/
- IMAP_DELETE_SUBFOLDERS=FALSE
- IMAP_FOLDER_PREFIX=
- IMAP_FOLDER_SEPARATOR=/
- IMAP_IDLE_TIMEOUT=30
- IMAP_LOGFILE=~/tmp/imap.%h
- IMAP_LOGLEVEL=0
- IMAP_MDNSENT_FLAG=$MdnSent
- IMAP_MIN_SIZE_ESTIMATE=0
- IMAP_REMOTE_UAL_ENABLED=TRUE
- IMAP_SEARCH_TIMEOUT=0
- IMAP_UAL_TRACE_LEVEL=0
- IMAP_X_NETSCAPE_URL=

The following table lists the IMAP4 options you can specify for individual clients.

Option	Description
IMAP_CAPABILITIES= *capabilities-list*	See the description in the <Xref_Color>"System-wide Configuration Options" section. Note that capabilities you specify here are *added* to those specified on a system-wide and per-user basis.
IMAP_MAILSTORE_HOST= *hostname*	Specifies the fully qualified domain name of the Scalix host to which the IMAP Server connects. Use this option when the IMAP4 Server does not reside on the same machine as the relevant Scalix message store.

mapi.cfg File

The mapi.cfg file sets parameters for all Scalix MAPI Client users and provides a way to customize some Outlook client functionality. This file is in the ~/nls/C/ directory on the Scalix Server if you are using Auto-upgrades (see the Scalix installation Guide for more information).

After a user logs into Outlook for the first time, the mapi.cfg file is automatically dowloaded to the local system from the Scalix Server.

The local **mapi.cfg** file is downloaded to the `C:\Documents and Settings\user\Local Settings\Application Data\Scalix\Scalix\MAPI\Profiles\profile_name\Scalix` directory.

The tables below list and describe the parameters you can configure in the **mapi.cfg** file.

Caution	If you modify any of these parameters then install (manually or automatically) an updated version of the MAPI service provide, the mapi.cfg file is replaced (overwrites the existing file) and the changes will be lost.

[AutoUpgrade] Parameters

Use this section to set Auto-upgrade options.

Parameter	Description
n	The mapi.cfg version number that is used to determine whether auto-upgrades occur. This number is also used to determine whether mapi.cfg is downloaded to update other administrative settings. NOTE: If the version number of the mapi.cfg file on the user system is *greater* than the version number of the mapi.cfg file on the server, Scalix does not upgrade Scalix Connect on the user system with the latest version of the MAPI service provider and/or update the mapi.cfg file.
SetupPath (8.2 to 9.1.1) HTTPSetupPath (9.2 and up)	The path to the shared drive/directory that contains the source Scalix Connect installation files. The `SetupPath` value must be a valid UNC path. The `HTTPSetupPath` must be a valid http:// adddress.
HTTPUpdateInstallMgr	This value is set to `1`. Do not modify this value.
HTTPUpgradeExemptList	Allows you to specify users that you do not want to upgrade to the latest version of Scalix Connect.
MinimumScalixVersion	The version number of the Scalix Connect dynamic link libraries.
ForwardInstallLogsTo	The administrator mailbox to which auto-upgrade results are sent. Enter `off` to disable error logging.
ForwardInstallLogsFrom	The text that displays in the From field of the Auto-install log message.
ForwardInstallLogsSubject	The Subject line of the e-mail that includes the auto-upgrade results.
UseLocalTimeVSGMT	Specify whether you want to use local time or Greenwhich Mean Time (GMT) to auto-upgrade users. Enter `1` to use local time, or `0` to use GMT.
UpgradeIntervalTimeCheck	The (metric) time at which Scalix polls client systems to verify whether they are using the latest version of Scalix Connect. For example, enter `8` to poll for Auto-upgrade status information at 8 am. Enter `22` to poll for information at 10 pm. Entering value of `24` or greater causes Scalix to poll for Auto-upgrade information in intervals (by seconds). For example, if you want to poll client systems every hour, enter `3600`.

[Startup] Parameters

Use this section to set startup options.

Parameter	Description
`AddressBookDownload ReminderInterval`	This option displays the number of days since you last downloaded a copy of the Address Book from the Scalix Server. Scalix Connect also reads the value of `PreviousABDownloadDate` in the registry key of `HKEY_LOCAL_MACHINE\SOFTWARE\SCALIX\MAPI`. Scalix Connect calculates the difference between the two dates. If the difference is greater than the value displayed in the `AddressBookDownloadReminderInterval` option, Scalix Connect displays a reminder to users to download a copy of the Address Book from the Scalix Server. To remind users to download Address Books monthly, set the value in the `AddressBookDownloadReminderInterval` option to `30`.
`AlwaysShowLogon`	The `AlwaysShowLogon=1` option specifies that the user is always prompted for a password at startup. If you enabled password storing at logon, you are not be prompted for a password.

[Addressing] Parameters

Addressing parameters affect the interpretation of Scalix addresses on messages.

Parameter	Description
InternetToOM	On an incoming message, Scalix Connect converts a Scalix address that includes a DDA (Domain Defined Attribute) type of RFC-822 to an address type of SMTP. For more information on DDAs, see the *Scalix Administration Guide*. Scalix Connect then uses the DDA for the revised address. For example, Scalix Connect replaces the Scalix address of a message such as: `chris/linux/dd.RFC-822=cwolfe@pwd.scalix.com` with an SMTP address such as `cwolfe@pwd.scalix.com`. You can override this behavior and keep the address as an Scalix type by including the setting `InternetToOM=1` in this section.
HPMEXTToSMTP	You can extend the conversion of Scalix addresses to SMTP addresses that include a DDA type of the form `HPMEXTn` by including the setting `HPMEXTToSMTP=1` in this section. The `InternetToOM=1` option takes precedence over the `HPMEXTToSMTP=1` option.

[Display] Parameters

The options in the `Display` section specify the following:

- the maximum number of items within a container
- which attributes are displayed
- the maximum line length in plain text messages
- how Internet Addresses are formatted

Settings in the [Display] section affect the presentation of Scalix addresses, for example, the displayed part of an address but not the underlying message address or type.

Parameter	Description
MaxContainerSize	The maximum number of messages that are listed on opening a folder can be configured using the MaxContainerSize setting. This setting can take an integer value between 20 and 32767. The default value is 32767. If the configured limit is exceeded then a warning message is displayed. Archiving (or auto-archiving) a folder that contains more than MaxContainerSize items causes the container-limit warning message to be displayed. Scalix Connect archive those items within the limit of 32767.
ShowMailnodes	Either the mailnode attributes or custom attributes can be displayed, but not both. You can set only one of the following options for attribute display. The format of each option is described in the following sections. The ShowMailnodes=1 option specifies that the mailnode attributes are displayed in message headers along with the name (Personal Name/OU1,OU2). This is useful when selecting similar entries from the directory. Without this setting you must scroll across the window to see the mailnode. The setting also applies to the display of addresses when either composing or reading a message. A way to resolve an unresolved address is to right-click on the address. This displays possible alternatives, which include the mailnode.
ShowCustomAttributes	The ShowCustomAttributes=1 option specifies that custom attributes, other than the mailnode details (Personal Name/OU1,OU2), are displayed in message headers. If you set this option, you must also set the UserDefinedAttributes option to specify the attributes to be displayed.
UserDefinedAttributes	UserDefinedAttributes=%(attr_tag)[%(attr_tag)] where attr_tag is the internal attribute tag for an Scalix Directory attribute type defined in the ~/sys/dir.attribs file. This tag can be either a predefined Scalix system attribute type (a numerical value) or a custom attribute type you have defined. Use the omshowatt -u command on the Scalix server to list the internal attribute tags. For example, specifying the line: UserDefinedAttributes=%(1)%(8)%(9) displays the Surname, Organization, and Country Code as the internal attribute tags for these Scalix attribute. The types are 1, 8, and 9, respectively, in the dir.attribs file. If you have defined a custom attribute type of JobTitle in the dir.attribs file, specifying the line: UserDefinedAttributes=%(1)%(JobTitle) displays the Surname and Job Title.

Parameter	Description
MaxContainerSize	The maximum number of messages that are listed on opening a folder can be configured using the `MaxContainerSize` setting. This setting can take an integer value between `20` and `32767`. The default value is `32767`. If the limit is exceeded, a warning message displays. Archiving (or auto-archiving) a folder that contains more than the `MaxContainerSize` value causes the container-limit warning message to display. Scalix Connect archives those items within the limit of `32767`.
LineLength	The maximum length of a line in a plain text message can be specified in the `Display` section. The format of the option is as follows: `LineLength=`*n* Specifies the maximum number of characters in each line of a plain text message, where *n* can be a value up to `80`. For example, `LineLength=60` ensures that the message text has no more than 60 characters per line. If the `Linelength=`*n* entry is missing or invalid then lines default to a maximum of 72 characters. To disable wrapping, add the entry `LineLength=0`.
`TabStops`	The `Tabstops=n` specifies the number of spaces for tab stops used by plain text messages, where *n* must not exceed the `LineLength` value or `20` (whichever is smaller). The default for *n* is `4` if the `Tabstops` setting is missing.
ShowComplete InternetAddress	The `ShowCompleteInternetAddress=1` option specifies that the entire Internet address is displayed. The `ShowCompleteInternetAddress=1` causes a Scalix address that contains a DDA type of RFC-822 to be displayed as an Internet address. For example, the address of an incoming message such as `chris/linux/dd.RFC-822=cwolfe@pwd.scalix.com` displays as `chris` (by default). If you set the value `ShowCompleteInternetAddress=1`, Scalix Connect displays the address as `cwolfe@pwd.scalix.com`. `ShowCompleteInternetAddress=1` takes precedence over `ShowMailnodes=1`. The behavior of `ShowCompleteInternetAddress` is also affected by settings in the `[Addressing]` section. A setting of `InternetToOM=1` prevents interpretation of the RFC-822 DDA as an Internet address and causes `ShowCompleteInternetAddress` to be ignored. `HPMEXTToSMTP=1` extends the behavior of `ShowCompleteInternetAddress` to HPMEXT DDAs.

Guidelines for the User Defined Attributes

For the UserDefinedAttributes option to be applied correctly, note the following guidelines:

The [Display] section must also include the following setting:

```
ShowCustomAttributes=1
```

The [Display] section must not include the setting ShowMailnodes=1 since this setting takes precedence over UserDefinedAttributes and results in an address of the following form:

```
Eric Smith / ou1, ou2
```

The UserDefinedAttributes setting can include up to ten entries, including a maximum of six custom attributes.

Any attribute specified in UserDefinedAttributes must also appear in the properties option of the Address Book. If the attribute is not one of the standard X.400 addressing fields or teletext equivalents (Scalix internal format Group 1 and 3), as displayed in the Name/Address Fields and the DDA pages, then you must add it to the custom attribute page through the mapi.cfg [Name Attributes] section.

In other words, any tags in UserDefinedAttributes outside the value ranges 1-23 and 51-68 must also appear in the [Name Attributes] section. For example:

```
[Name Attributes]
Heading=Custom
1=Phone:, 116
2=My Own Data:, myown
[Display]
ShowMailnodes=0
ShowCustomAttributes=1
UserDefinedAttributes=%(2)%(1)%(116)%(myown)
```

[Directories] Parameters

The option in the [Directories] section enables additional directories from the server to be included. The server default Directory is always opened by the MAPI Address Book provider.

Parameter	Description
n=directory name	Specifies an additional Directory, where: *n* is a consecutive sequence number (1-20). *directory name* is the name of the additional directory to be included. Directory names are case-sensitive. For example, to include Directories for your Sales and Overseas departments, specify the following lines: [Directories] 1=SALES 2=OVERSEAS Make sure these entries are case sensitive, for example if you have the directory `MYOWNONE Shared LOCAL DB config update read modify-self` (as shown by `omlistdirs`), the entry in this section must be: 1=MYOWNONE You also need to add these directories to the CDA server (`omaddcda`) and then run `omexeccda`. Add the `-d <directoryname>` option to enable type-down searching on this directory.

[Name Attributes] Parameters

The option in the Name [Attributes] section enables additional attributes from the Scalix Directory to be included as a name/address properties page and as the Search page of the MAPI client.

Parameter	Description
heading=*text* *n=label,tag*	This specifies the additional attributes for the Properties and Search pages, where: text is the text used as the page Tab heading. You can specify up to 16 characters. n is the sequence number for the custom attribute. You can specify up to six attributes (1-6). label is the text for the label displayed for the attribute on the page. You can specify up to 24 characters. tag is the numeric value of the internal attribute tag for the corresponding Scalix Directory attribute. To see the list of available tags, use the omshowatt -u command. For example, to include three additional custom attributes, specify the following lines: [Name Attributes] heading = Custom 1=Job Title:, 111 2=Department:, 115 3=Phone:, 116

[PAW] Parameters

The [PAW] (Personal Assistant Wizard) section in mapi.cfg has the following options:

Parameter	Description
URL	The URL pointing to the web server for the user, for example: URL=http://zaphod.pwd.scalix.com This option is required for PAW to be available to the user.
AutoLogon	The automatic logon option uses either `1` or `0` as a value. `1` bypasses the web-based logon page and logs on automatically to the PAW home page. `0` does not bypass the web-based logon page and you must enter your username and password. This option is not required. PAW is still available to a user even if this option is not set.
Profile	The profile option determines the language for the PAW application at start-up. For example: Prof=PAW-ENGLISH This option is not required. PAW is still available to a user even if this option is not set.

Restrictions

Before Scalix Connect includes PAW as a menu option (Tools > Personal Administration Wizard), Scalix Connect checks the following requirements:

- A valid Scalix Server profile.
- A valid **mapi.cfg** file with the `[PAW]` section included in the configuration file.

- If the [PAW] section exists within the **mapi.cfg** file, Scalix Connect checks that the URL is defined for the PAW server.

The PAW option in the Outlook Tools menu is unavailable if any of these requirements are missing or invalid.

User-Specific Configuration Options

A subset of the options used in the general.cfg file can be specified for individual users. The options for each user are held in files with the name of that user's Scalix ID number in the directory ~/sys/user.cfg.

For example, if the Scalix user Chris Wolf/ny,hq,mis has a Scalix ID of 103, then options specific to him will be in the file ~/sys/user.cfg/103.

The user.cfg directory does not exist by default and must be created. It must be owned by the user Scalix with permissions of 555 (dr-xr-xr-x). User files within this directory must be owned by the user Scalix with permissions of 444 (-r--r--r--).

See the descriptions in the <Xref_Color>"System-wide Configuration Options" section for the following options:

- IMAP_AUTOMATIC_MDN=FALSE
- IMAP_BB_FOLDER_PREFIX=#bb
- IMAP_BB_FOLDER_SEPARATOR=/
- IMAP_DELETE_SUBFOLDERS=FALSE
- IMAP_FOLDER_PREFIX=
- IMAP_FOLDER_SEPARATOR=/
- IMAP_IDLE_TIMEOUT=30
- IMAP_LOGLEVEL=0
- IMAP_MDNSENT_FLAG=$MdnSent
- IMAP_MIN_SIZE_ESTIMATE=0
- IMAP_SEARCH_TIMEOUT=0
- IMAP_X_NETSCAPE_URL=

IMAP Client User-specific Options

Option	Description
IMAP_CAPABILITIES= *capabilities-list*	See the description in the <Xref_Color>"System-wide Configuration Options" section. Note that capabilities you specify here are *added* to those specified on a system-wide and client-wide basis.
IMAP_LOGFILE=~/tmp/imap.%h	See the description in the <Xref_Color>"System-wide Configuration Options" section. Note that, if logging has been enabled in the system-wide or client-specific configuration file, this option has no effect.

UAL Client Interface User-specific Options

Option	Description
UAL_DEAD_TIMEOUT=*number_of_minutes*	Specifies the amount of time that Scalix will wait for the next UAL command of any type (as opposed to UAL_IDLE_TIMEOUT, which is triggered by active commands only) from a UAL client before assuming a timeout. Once the timeout period has been reached, Scalix assumes the connection to the client has been lost and signs off the user of the UAL client. If a timeout period is not specified, Scalix assumes the connection to the UAL client is good regardless of how long it has been waiting for another command. For serial connections, UAL_DEAD_TIMEOUT is overridden by UAL_SERIAL_DEAD_TIMEOUT. For local UAL clients, UAL_DEAD_TIMEOUT is overridden by UAL_LOCAL_DEAD_TIMEOUT.
UAL_DIR_LIST_SORT_ORDER=*list_of_internal_attributes*	Specifies the order in which Directory attributes are sorted. The order is specified as a list of internal attribute names with each attribute separated by a /. The internal attribute names, which are numbers for the core Scalix attributes, are listed using the command omshowatt -u.
UAL_DIR_MOD_FULL_NAME=TRUE	Specifies that Full Name Checking is always done on the UAL_CHKLST, UAL_CHKNAM, UAL_DELENT and UAL_MODENT commands.
UAL_DISALLOW_AUTO_PASSWORD=TRUE	If set, a client cannot sign on to Scalix if the client has explicitly indicated that its password was obtained from a configuration file rather than having been entered interactively by a user. See also UAL_DISALLOW_NON_USER_PASSWORD. The user-specific setting of this option overrides the system-wide setting. Note that this mechanism is not intended to provide a secure indication.
UAL_DISALLOW_NON_USER_PASSWORD=TRUE	If set, a client cannot sign on to Scalix if the client has *not* explicitly indicated that its password was obtained interactively from a user. See also UAL_DISALLOW_AUTO_PASSWORD. Note that this option will only work with clients that supply the "password origination status". If a client does not support this element, then it will not be able to sign on even if the password is actually entered interactively by the user. The user-specific setting of this option overrides the system-wide setting. Note that this mechanism is not intended to provide a secure indication.

Option	Description
UAL_DL_SIZE_LIMIT= *no_of_kilobytes*	Sets the Distribution List area size limit. in kilobytes. A value of zero (0) means no size limit. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_FC_SIZE_LIMIT= *no_of_kilobytes*	Sets the Filing Cabinet size limit. The value is set in kilobytes. A value of zero means no size limit. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_IDLE_SHUTDELAY= *umber_of_minutes*	Specifies the additional delay in shutting down a UAL client connection that has timed out. `UAL_IDLE_SHUTDELAY` is used with `UAL_IDLE_TIMEOUT`. See also `UAL_IDLE_TIMEOUT` For serial connections, `UAL_IDLE_SHUTDELAY` is overridden by `UAL_SERIAL_IDLE_SHUTDELAY`. For local UAL clients, `UAL_IDLE_SHUTDELAY` is overridden by `UAL_LOCAL_IDLE_SHUTDELAY`.

Option	Description
UAL_IDLE_TIMEOUT= *number_of_minutes*	Specifies the amount of time that Scalix will wait for the next "active" UAL command from a UAL client before assuming a timeout (`PREPARE MESSAGE`, `ATTACH ITEM` are examples of active UAL commands, and `NEW MESSAGES` and `LIST ACK` are examples of passive UAL commands). Once the timeout period has been reached, Scalix assumes the connection to the client has been lost and signs off the user of the UAL client. (The sign off can be delayed using `UAL_IDLE_SHUTDELAY`.) If a timeout period is not specified, Scalix assumes the connection to the UAL client is good regardless of how long it has been waiting for another command. `UAL_IDLE_TIMEOUT` is used with `UAL_IDLE_SHUTDELAY`. For example: If `UAL_IDLE_TIMEOUT` is set to 30 minutes, and `UAL_IDLE_SHUTDELAY` is not set, the client is disconnected from the Server 30 minutes after the last active UAL command was issued. If `UAL_IDLE_TIMEOUT` is set to 30 minutes, and `UAL_IDLE_SHUTDELAY` is set to 10 minutes, 30 minutes after the last active UAL command is issued, the client displays a dialog box asking if the user wants to retain the connection. This dialog box is displayed for up to the 10 minutes specified by `UAL_IDLE_SHUTDELAY`. If the user responds within this time with a Yes, that is considered an active UAL command, and the `TIMEOUT` countdown restarts from the beginning. If the user responds with a No, the connection is closed. If the user does not respond within the 10 minutes, the connection is closed. For serial connections, `UAL_IDLE_TIMEOUT` is overridden by `UAL_SERIAL_IDLE_TIMEOUT`. For local UAL clients, `UAL_IDLE_TIMEOUT` is overridden by `UAL_LOCAL_IDLE_TIMEOUT`.
SpUAL_INTRAY_SIZE_LIMIT= *no_of_kilobytes*	Sets the In Tray size limit in kilobytes. A value of zero (0) means no size limit. Set Message Store size limits using the `omlimit` command. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_ISO7_FROM_HOST=*language*	This option is the same as `UAL_ISO7_HOST` except that the character set conversion only occurs when text is passed from Scalix to the client and not when it is passed back to the Server.

Option	Description
UAL_ISO7_HOST=*language*	This option allows clients to interoperate with a Scalix Message Store containing IA5 text that uses the ISO-7 extensions. If set, activates the option `UAL_ISO7_`*language* for a client using language. IA5 text with the ISO-7 extensions are converted to the ISO8859/1 character set when downloaded to or displayed by the client and conversely, the ISO8859/1 characters are mapped back into IA5 with ISO-7 extensions when entering the Scalix system from the client. Only one instance of this option can be used and the language string must match a string in the `~/sys/LangMap` file. See also `UAL_ISO7_FROM_HOST, UAL_ISO7_TO_HOST,` and `SR_ISO7_HOST`.
UAL_ISO7_TO_HOST=*language*	This option is the same as `UAL_ISO7_HOST` except that the character set conversion only occurs when text is passed from the client to Scalix and not when it is passed back to the client.
UAL_LOCAL_DEAD_TIMEOUT=*number_of_minutes*	Specifies the amount of time that Scalix will wait for the next UAL command of any type (as opposed to `UAL_LOCAL_IDLE_TIMEOUT`, which is triggered by active commands only) from a local UAL client before assuming a timeout. Once the timeout period has been reached, Scalix assumes the connection to the client has been lost and signs off the user of the UAL client. If a timeout period is not specified, Scalix assumes the connection to the local UAL client is good regardless of how long it has been waiting for another command. `UAL_LOCAL_DEAD_TIMEOUT` overrides `UAL_DEAD_TIMEOUT`. To remove a timeout for local UAL clients that was set using `UAL_DEAD_TIMEOUT,` set `UAL_LOCAL_DEAD_TIMEOUT` to 0.
UAL_LOCAL_IDLE_SHUTDELAY=*number_of_minutes*	Specifies the additional delay in shutting down a local UAL client connection that has timed out. `UAL_LOCAL_IDLE_SHUTDELAY` is used with `UAL_LOCAL_IDLE_TIMEOUT`.

Option	Description
UAL_LOCAL_IDLE_TIMEOUT= *number_of_minutes*	Specifies the amount of time that Scalix will wait for the next "active" UAL command from a local UAL client before assuming a timeout (`PREPARE MESSAGE`, `ATTACH ITEM` are examples of active UAL commands, and `NEW MESSAGES` and `LIST ACL` are examples of passive UAL commands). Once the timeout period has been reached, Scalix assumes the serial connection to the client has been lost and signs off the user of the UAL client. (The sign off can be delayed using `UAL_LOCAL_IDLE_SHUTDELAY`.) If a timeout period is not specified, Scalix assumes the connection to the UAL client is good regardless of how long it has been waiting for another command. `UAL_LOCAL_IDLE_TIMEOUT` is used with `UAL_LOCAL_IDLE_SHUTDELAY`. `UAL_LOCAL_IDLE_TIMEOUT` overrides `UAL_IDLE_TIMEOUT`. To remove a timeout for local UAL clients that was set using `UAL_IDLE_TIMEOUT`, set `UAL_LOCAL_IDLE_TIMEOUT` to 0.
UAL_LOCAL_IGNORE_PASSWORD= TRUE or FALSE	Specifies whether a password check is made during sign on. Set the option to `TRUE` to remove the password entry stage from the sign on process. Set the option to `FALSE` to add the stage back into the sign on process if it has been removed by setting `UAL_LOCAL_IGNORE_PASSWORD` in the `general.cfg` file. The sign on will succeed only if the user has logged in using their Scalix mailbox Linux login, and if the user is using a local UAL Client.
UAL_MSTORE_SIZE_LIMIT= *no_of_kilobytes*	Sets the overall message store size limit. The value is set in kilobytes. A value of zero means no size limit. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_NMP_DELAY= *number_of_milliseconds*	Specifies the time delay before returning a reply to a command from a UAL client using a TCP/IP Named Pipes connection. By default, there is no time delay, but this can mean the receiving client system can "miss" the reply when it is sent. To overcome this problem set the delay to between 10 and 100 milliseconds.
UAL_NO_DESIGNATE_SIGNON= TRUE or FALSE	Specifies whether the designate sign on feature is used. Set the value to `TRUE` to remove the designate sign on feature. Set the value to `FALSE` to add the designate sign on feature if it has been removed by setting `UAL_NO_DESIGNATE_SIGNON` in the `general.cfg` file.
UAL_NO_WB_EMPTY=TRUE	Stops a user's Waste Basket being emptied when the user has finished using a UAL client and signs off. If this option is set, use the command omtidyu or omtidyallu to ensure Waste Baskets continue to be emptied regularly.

Option	Description
UAL_PASSWORD_AGED=*IGNORE WARN or ERROR*	This option determines the effect of an expired password on a user signing on to Scalix through a client. The default value is ERROR. If the user's password has expired, an error is generated when the user attempts to signon and the signon fails. The signon can only succeed when a valid new password is supplied. If the value is set to WARN and the user's password has expired, the user can sign on using the expired password but a warning message is placed in their In Tray stating that their password has expired and should be changed immediately. (This message appears in the In Tray for the first signon of the day.) If the value is set to IGNORE any user password expiry condition is ignored (a Scalix user will be allowed to signon even if their password has expired.)
UAL_PEND_SIZE_LIMIT= *no_of_kilobytes*	Sets the Pending Tray size limit in kilobytes. A value of zero (0) means no size limit. Set Message Store size limits using the omlimit command. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.
UAL_PWD_WARNING_DAYS=*days*	Activates the mechanism to generate advisory messages to users whose mailbox passwords are due to expire within the period specified by days. The warning message appears as a new message in the user's In Tray for the first signon of the day. Use this option if clients are being used that do not recognize the password expired signon error. These clients cannot signon successfully once the user's password has expired.
UAL_SCK_DELAY= *number_of_milliseconds*	Specifies the time delay before returning a reply to a command from a UAL client using a TCP/IP Sockets connection. By default, there is no time delay, but this can mean the receiving client system can "miss" the reply when it is sent. To overcome this problem set the delay to between 10 and 100 milliseconds.
UAL_SERIAL_DEAD_TIMEOUT= *number_of_minutes*	Specifies the amount of time that Scalix will wait for the next UAL command of any type (as opposed to UAL_SERIAL_IDLE_TIMEOUT, which is triggered by active commands only) from a UAL client using a serial connection before assuming a timeout. Once the timeout period has been reached, Scalix assumes the connection to the client has been lost and signs off the user of the UAL client. If a timeout period is not specified, Scalix assumes the serial connection to the UAL client is good regardless of how long it has been waiting for another command. UAL_SERIAL_DEAD_TIMEOUT overrides UAL_DEAD_TIMEOUT. To remove a timeout for UAL clients that was set using UAL_DEAD_TIMEOUT, set UAL_SERIAL_DEAD_TIMEOUT to 0. This removes the timeout for all UAL clients using a serial connection.

Option	Description
UAL_SERIAL_IDLE_SHUTDELAY= *number_of_minutes*	Specifies the additional delay in shutting down a UAL client serial connection that has timed out. UAL_SERIAL_IDLE_SHUTDELAY is used with UAL_SERIAL_IDLE_TIMEOUT.
UAL_SERIAL_IDLE_TIMEOUT= *number_of_minutes*	Specifies the amount of time that Scalix will wait for the next "active" UAL command from a UAL client using a serial connection before assuming a timeout (PREPARE MESSAGE, ATTACH ITEM are examples of active UAL commands, and NEW MESSAGES and LIST ACL are examples of passive UAL commands). Once the timeout period has been reached, Scalix assumes the serial connection to the client has been lost and signs off the user of the UAL client. (The sign off can be delayed using UAL_SERIAL_IDLE_SHUTDELAY.) If a timeout period is not specified, Scalix assumes the connection to the UAL client is good regardless of how long it has been waiting for another command. UAL_SERIAL_IDLE_TIMEOUT is used with UAL_SERIAL_IDLE_SHUTDELAY. UAL_SERIAL_IDLE_TIMEOUT overrides UAL_IDLE_TIMEOUT. To remove a timeout for UAL clients that was set using UAL_IDLE_TIMEOUT, set UAL_SERIAL_IDLE_TIMEOUT to 0. This removes the timeout for all UAL clients using a serial connection.
UAL_SIGNON_ALIAS=YES, ONLY or NONE	Specifies whether aliases are used for sign on. Any UAL_SIGNON_ALIAS entries in user.cfg take precedence over the UAL_SIGNON_ALIAS entry in general.cfg. (This enables you to set a default use of aliases in general.cfg and then set overrides for specific users in user.cfg.) The YES option means aliases can be used to sign on with--users can also continue to use their Personal Name if they want to. The ONLY option means the aliases only can be used to sign on with--the Personal Name cannot be used any more. The NONE option means aliases are not used. UAL_SIGNON_ALIAS is used with UAL_SIGNON_ALIAS_CONFIG and UAL_USE_SIGNON_ALIAS.
UAL_SIZE_ON_RECEIPT=FALSE	Specifies whether a user whose message store components exceed their configured limits is prevented from receiving messages. When this option is set to FALSE (the default), users are not prevented from receiving messages even if the size of their message store component is greater than its configured limit.

Option	Description
UAL_SIZE_ON_SEND=FALSE	Specifies whether a user whose message store components exceed their configured limits is prevented from sending messages. When this option is set to TRUE, then message delivery rules can be implemented that limit a user's ability to send messages. These rules utilize the OMLIMIT-EXCEEDED message attribute filter. When this option is set to FALSE (the default), then rules based on the OMLIMIT-EXCEEDED filter have no effect.
UAL_TRACE_FILE= *file_specification*	The name stub of the file to which UAL trace information is logged (UAL logging must be enabled). %p in the *file_specification* is replaced with the PID of the UAL process, %s by the notification session-ID, and %u by the Scalix UID. A leading ~ represents the Scalix home directory. Note that the existing log file is overwritten. Without a leading ~ or / character, the file(s) are created in the ~scalix/tmp directory. UAL trace output is enabled by using the UAL_TRACE_LEVEL option. The default value is OM%u. The substitutions in the log file name allow log files to be created on a per-ual-process, per-notif-session, or per-user basis. This allows MAPI and Scalix Web Access sessions that use concurrent UAL sessions to be traced without any loss of data. UAL client session must be restarted to enable the changes to this option. Example: UAL_TRACE_FILE=ual.%u.%p creates log files in the ~scalix/tmp directory with a stub of ual.*user_id.pid*, For example: ual.102.1773U.log ual.102.1773U.f0001 Example: UAL_TRACE_FILE=/tmp/ual-logs/%u.%p creates log files in the /tmp/ual-logs directory with a stub of *user_id.pid*, For example: /tmp/ual-logs/102.1773U.log /tmp/ual-logs/102.1773U.f0001 In this example, the /tmp/ual-logs directory must be created before any trace files can be written. See UAL_TRACE_LEVEL for more information.

Option	Description
UAL_TRACE_LEVEL=*trace_level*	Activates UAL Client Interface tracing. The trace files are placed in the `~/tmp` directory. If this directory cannot be found, they are placed in the `/tmp` directory. File names begin with OM*user-no*, where *user-no* is the Scalix user number, and end according to the trace level set. If you require several different kinds of trace information, add the numbers for the levels you require and set the entry to the total. 0: No tracing. The default. 1: Raw (unformatted) command/reply tracing (file name: *name*`N.trc`). 2: Command statistics. 4: Message Store file name mapping. No trace file. The subject of an item listed or displayed in the client is replaced by its corresponding Message Store file name. 8: Full tracing of command/reply and file transfer data. This can be used to rerun a session (file name: *name*`U.log` and *name*`U.f`*nnnn*). 16: Raw (unformatted) command/reply tracing and file transfer data (file name: *name*`N.trc`). Also use this entry to set Event Log logging on the Server for the client. Set the entry to the required Event Log logging level multiplied by 100.
UAL_LINUX_PASSWORD=TRUE or FALSE	Specifies whether the user uses their Linux password instead of their Scalix password when signing on. `TRUE` sets Scalix to use the Linux password, and `FALSE` (the default) sets Scalix to use the Scalix password.
UAL_USE_SIGNON_ALIAS=FALSE or TRUE	Specifies whether the alias is used after sign on. If you set `UAL_USE_SIGNON_ALIAS` to `FALSE`, the UAL client reverts to using the user's Personal Name for the remaining time the user is signed on (the alias or Personal Name is used on the "Creator" part of a message). If you set `UAL_USE_SIGNON_ALIAS` to `TRUE`, the alias is used for the remaining time the user is signed on. `UAL_USE_SIGNON_ALIAS` is used with `UAL_SIGNON_ALIAS` and `UAL_SIGNON_ALIAS_CONFIG`.
UAL_WB_SIZE_LIMIT= *no_of_kilobytes*	Sets the Waste Basket size limit in kilobytes. A value of zero (0) means no size limit. Set Message Store size limits using the `omlimit` command. Once you configure a limit, you must then configure the sanctions to be applied to those users who exceed it. If you do not configure sanctions, users can exceed their limits without effect.

Offline Folder Synchronization Options (Outlook Clients)

Option	Description
OFS_ENABLED=FALSE	Specifies whether folder synchronization is enabled on the Scalix Server. The default is FALSE. If this option is set to TRUE in `general.cfg`, it can be overridden on a per-user basis by setting it to FALSE in the relevant user-specific configuration files.
OFS_LOG_SIZE_LIMIT=*kilobytes*	Specifies, in kilobytes, the maximum size of the folder synchronization change log. Set a value between 20 and 10,000 KB. The default is 100 KB. When the size of a change log exceeds this value, the older entries can be deleted when the change log file is compacted. Use this option to prevent change logs from growing too large when logging is activated but synchronization is not taking place for any reason. Take care when you set this value, since removal of any valid entries will cause the entire folder to be resynchronized. A value you set in `general.cfg` can be overridden on a per-user basis by setting a lower value for this option in the relevant user-specific configuration file.
OFS_LOG_AGE_LIMIT=*days*	Specifies, in days, the maximum age of entries in the folder synchronization change log. Set a value between 1 and 18,000 days. The default is 90 days. When the age of a change log entry exceeds this value, it can be deleted when the change log file is compacted. Use this option to prevent change logs from growing too large when logging is activated but synchronization is not taking place for any reason. Take care when you set this value, since removal of any valid entries will cause the entire folder to be resynchronized. A value you set in `general.cfg` can be overridden on a per-user basis by setting a lower value for this option in the relevant user-specific configuration file.

Chapter 25

Scalix Command Line Reference Guide

This publication summarizes the following collections of Scalix CLI commands.

Introduction

After you log into your Scalix server , you can enter Scalix commands directly in a Linux terminal window, using the full Scalix command line interface. Scalix commands are constructed with a prefix of "om" followed by extensions that define the action and the object. For example, the command to add a user is **omaddu** (followed by the needed extensions).

Because interpretations of command characters vary per shell program, you might need to use escape sequences for some command elements. For example, if you enter parentheses while using the command **omsearch**, some shell programs might require that the parentheses be escaped with backslashes (\) before the shell program can interpret the command correctly.

Note	Some Scalix commands can be used only by a Scalix Administrator with system administration privileges (for example, a root user).

The following table lists the location of additional information (manual pages) about command formats and matching rules not covered in this chapter.

Command	Use
omaddress	Enter O/R Address, mailnodes, and pattern matching rules
omattribs	Attribute input and output formats
omdiratt	Formatting of attribute definition files
scalix	Services, queues, daemons, and commands

Starting with the following page, this publication provides a complete catalog of all Scalix CLI commands, listing them in common categories, and providing a simple description of how each command is used. For more information about syntax and extensions, see the man page for each command.

Access Control List Commands

Command	Uses
omaddacl	Add an Access Control List
omaddacln	Add capabilities for users to an Access Control List
omchkacln	Check capabilities of a user in an Access Control List
omdelacl	Delete an Access Control List
omdelacln	Delete capabilities for users from an Access Control List
ommodacln	Modify capabilities for users in an Access Control List
omshowacl	Show the contents of an Access Control List

Audit Log Commands

Command	Uses
omconfaud	Configure Audit Log logging levels
omshowaud	Show Audit Log logging levels

Bulletin Board (Public Folder) Commands

Command	Uses
omaddbbsa	Add a Bulletin Board Synchronization agreement
omdelbbsa	Delete a Bulletin Board Synchronization agreement
omlistbbsa	List Bulletin Board Synchronization agreements
ommodbbsa	Modify a Bulletin Board Synchronization agreement

Client Directory Access (CDA) Commands

Command	Uses
omaddcda	Add a Directory to the CDA Server configuration
omdelcda	Delete a Directory from the CDA Server configuration
omexeccda	Force the CDA Server to process a Directory immediately
ommodcda	Modify the CDA Server configuration for a Directory
omshowcda	Show the CDA Server configuration for a Directory

Configuration and Installation Commands

Command	Uses
omcptree	Copy or refresh a Directory hierarchy
omdelom	Delete a Scalix instance
ommakeom	Make a Scalix instance
ompatchom	Update a Scalix instance
omredirtcp	Redirect socket connections to the correct Scalix system in a multi-system environment

Directory Commands

Command	Uses
omaddent	Add one or more entries to a Directory
omdelent	Delete one or more entries from a Directory
omdiropt	Optimize a Directory
omdoptall	Optimize all Directories
omfmtent	Format Directory and address attributes
omlistdirs	List Directories

Command	Uses
ommoddir	Modify a Directory
ommodent	Modify a Directory entry
omremdir	Delete a Directory
omsearch	Search a Directory
omshowatt	Show available attribute types

Directory Relay Server Commands

Command	Uses
omresetmn	Reset a mailnode mapping file
omaddmnmp	Add entry to a mailnode mapping file
ommodmnmp	Modify entry in a mailnode mapping file
omdelmnmp	Delete entry in a mailnode mapping file
omshowmnmp	List entries in a mailnode mapping file

Directory Synchronization Commands

Support for multiple Scalix servers and directory synchronization is available only in Scalix Enterprise Edition. For more information, see "About Scalix Product Editions".

Command	Uses
omaddds	Add a Directory Synchronization agreement
omdelds	Delete a Directory Synchronization agreement
omlistds	List Directory Synchronization agreements
ommodds	Modify a Directory Synchronization agreement
omresyncds	Resynchronize a Directory
omshowds	Show details of a Directory Synchronization agreement

Error Manager Commands

Command	Uses
omconfenu	Configure an Error Manager
omshowenu	Show the address of the Error Manager

Event Log Commands

Command	Uses
omconflvl	Configure Event Log logging levels
omshowlog	Show the Event Log
omshowlvl	Show Event Log logging levels

Internet Address Commands

Command	Uses
omaddiam	Add entry to the Internet address mapping file
omdeliam	Delete an entry from the Internet address mapping file
omgeniamods	Generate a script to modify Internet addresses.
ommodiam	Modify a mapping between OR address and Internet address
ompreviewia	Preview the automatically generated Internet address
omshowiam	List mappings between OR addresses and Internet addresses

Internet Mail Gateway Commands

Command	Uses
omconfux	Configure the Internet Mail Gateway
omshowux	Show the configuration of the Internet Mail Gateway

LDAP Commands

Command	Uses
omldapadd	Add one or more entries to an LDAP Directory
omldapdelete	Delete one or more entries from an LDAP Directory
omldapmodify	Modify an LDAP Directory entry
omldapmoddn	Modify the DN of an LDAP entry
omldapsearch	Search an LDAP Directory

Mailbox Access Commands

Command	Uses
omdelete	Delete a message
omlist	List messages
omlogoff	Terminate an `omlogon` connection to Scalix
omlogon	Obtain a connection to Scalix
omnew	List newly arrived messages
omread	Read a message
omsend	Send a message

Mailnode Commands

Command	Uses
omaddmn	Add a mailnode
omdelmn	Delete one or more mailnodes
ommodmn	Modify a mailnode
omshowmn	List local mailnodes

Message Store Commands

Command	Uses
omcontain	Manipulate containers in the Message Store
omcpinu	Copy a user's Message Store data from a file
omcpoutu	Copy a user's Message Store data to a file
omdosur	Create a data file for restoring a single user
omdref	Convert a Scalix DirectRef into a readable description of the item represented, including Message Store item hierarchy
omdumpis	Write Item Structure database to standard output
omgetsur	Get files from an archive
omlimit	Set Message Store size limits globally or for a user
omnewis	Create an empty database
omprepsur	List files required for single user restore
omscan	Scan, report, and repair Scalix data inconsistencies
omshowis	Display the date `omupdtis` was last run
omsnoop	Report on potential Message Store conversion problems
omsuspend	Halt all client activity temporarily
omtidyallu	Delete items from the Message Store
omtidyu	Delete items from the Message Store for an individual and search nested folders for an item to delete
omupdtis	Read Item Structure log entries and update the database
tfbrowse	Convert between Scalix transaction file format and textual format

Miscellaneous Commands

Command	Uses
ombconv	Convert a numeric value into a variety of numeric bases

Command	Uses
ombprint	Print messages in batch mode
omenquire	Enquire about Scalix system status and report the results
omsolve	Display solutions to an error message

Public Distribution List (PDL) Commands

Command	Uses
omaddpdl	Add a Public Distribution List
omaddpdln	Add an entry to a Public Distribution List
omdelpdl	Delete one or more Public Distribution Lists
omdelpdln	Delete one or more entries from a Public Distribution List
ommodpdl	Modify a Public Distribution List
ommodpdln	Modify Public Distribution List entries
omshowpdl	List Public Distribution Lists
omshowpdln	List entries in a Public Distribution List
omaddaci	Add an Access Control Information member
omchkaci	Check Access Control Information capabilities for a user
omdelaci	Delete an Access Control Information member
ommodaci	Modify an Access Control Information member
omshowaci	Show the contents of Access Control Information

Public Folder Commands

Public folders can be accessed only by Premium users, using an IMAP client or SWA. For more information, see "About Scalix Product Editions".

Command	Uses
omaddbb	Add a top-level Public Folder
omdelbb	Delete a top-level Public Folder
omlistbbs	List top-level Public Folders
ommaintbb	Maintain top-level Public Folders
ommodbb	Modify the subject of a top-level Public Folder
omshowbb	Show details of a top-level Public Folder
omaddbbsa	Add a Public Folder Synchronization agreement
omdelbbsa	Delete a Public Folder Synchronization agreement
omlistbbsa	List Public Folder Synchronization agreements
ommodbbsa	Modify a Public Folder Synchronization agreement

Routing Table Commands

Command	Uses
omaddrt	Add a route
omdelrt	Delete a route
ommodrt	Modify a route
omshowrt	List routes and show how an address is routed

Service, Queue and Daemon Commands

Command	Uses
omisoff	Check Scalix services are off

Command	Uses
omoff	Stop one or more services
omon	Start one or more services
omrc	Start Scalix
omreset	Reset status of services or remove Scalix
omresub	Resubmit messages
omresubdmp	Resubmit messages processed by the Archive Server
omsetsvc	Display the status of a service in detail; configure auxiliary processes
omshut	Stop Scalix
omstat	List Scalix daemons

System Configuration and Maintenance Commands

Command	Uses
omcheck	Check Scalix file permissions and ownership
omcnvinst	Configure converter, language, and character set files
ommon	Monitor the operation of Scalix
omstat	Show the status of the system
omvers	List version numbers of all binaries and scripts

User Entry and Management Commands

Command	Uses
omaddu	Add a user
omadmidp	Configure system IDs for use by Scalix users
omconfpwd	Configure password controls

Command	Uses
omdelu	Delete one or more users
ommoddl	Modify distribution list entries and auto-action addresses
ommodu	Modify a user
omshowpwd	Show password controls
omshowu	List users or display details about a specific user