Contents:

Search: Enter Word(s) to find

Submit

Free Trial Updates

License Prices

Products Downloads

Support Company

Contents:

Installing and Initial Setup of the DMail Package

(To: Table of Contents)

The parts of the package are:

DSMTP - SMTP mail server

DPOP - POP3 mail server

DList - email list server

DWatch - monitor for the above 3 servers

Pre-Installation

(To: install instructions)

Before installing DMail you need to think about:

- **Servers already on your system**

  Before DPOP is started the original popper must be disabled and the same applies for any SMTP server that you have already. You must do this manually before letting the DMSetup installation wizard start.

  (On Unix type systems if you are running sendmail this is normally done by editing inetd.conf and then sending the inet process a HUP signal to get it to reload. The DMSetup wizard will perform these steps for you if requested.)

  Running on non-standard ports:
  Alternatively DPOP and DSMTP can be installed and not started (DMSetup asks you if you want the servers started). Later you can start them manually after first changing the ports that they operate on so that they do not interfere with the operation of another popper or SMTP server. For details see the config settings smtp_port and pop_port

Installation

The DMSetup setup application wizard will guide you through the initial setup and installation of the DMail package which includes DSMTP, DPOP and DList. It can also be used to simplify upgrading to a newer version of DMail or for removing the DMail applications. It will automatically determine which package you have downloaded and whether you are installing just DPOP or the complete package including DMail, DPOP and DList.

A note on virtual domains:
If you are planning to add many domains, then we suggest that you pick one of these domains to be your main domain. Use DMSetup to add only this domain, get it working such that you can send and receive email and only then set about adding the other domains. The other domains that you add we refer to in this manual as 'virtual domains' although there is very little difference between them and your main domain. Virtual domains are created in dmail.conf with 'vdomain' lines, where as your first or 'main' domain is created with a host_domain setting.

DMSetup will create the required configuration files for all the server products (almost everything is in dmail.conf). These configuration files will be tailored to your system and selected preferences. Nearly all the settings can be changed later but it is generally best to use the DMSetup program at least once to get your main configuration options correct. To install the DMail products you proceed as follows:

On Unix

1. Log in as root
2. Download the DMail distribution set to a temporary directory
   (e.g. root/chris/temp/dm119_linux.tar.Z)
3. Unpack the files with the commands like:
4. **uncompress dm119_linux.tar**

   tar -xvf dm119_linux.tar
5. Change directory **cd dmtemp**
6. Use command **./dmsetup** to run the installation wizard
7. Answer the questions the wizard asks.
8. Test the DSMTP/DPOP/DList mail servers you have installed.

On Windows NT

1. Download the DMail distribution set into a temporary directory
2. At completion of download the DMSetup wizard will be started automatically
3. Answer the questions the DMSetup wizard asks.
4. Test the DSMTP/DPOP/DList mail servers you have installed.

Note: If NT_User passwords are used then DPOP must be run with enough privilege to allow password checking. i.e. from an account with the "act as part of operating system" right set.

The DMSetup Wizard

This is an initial setup program which runs from the self-extracting zip file on Windows NT and the ./dmsetup command on Unix.

It asks a series of setup questions, most of which have default responses (given in square brackets), which you can select by simply pressing a carriage return. All of the responses that you give to DMSetup can be changed later on, after installation, so it is often a good idea just to accept the defaults and then change the ones you need to be different once you are more familiar with the package.

CLick here for Notes on DMSetup Questions

DMSetup creates an initial configuration file, dmail.conf, based on the responses that you give to it's

questions. It also copies the DMail files to the various locations, including the configuration file, dmail.conf, to the system directory.

On both Windows and Unix it will try by default to create a directory called 'dmail' and to point all of the server's path settings to this directory, e.g. it will point dsmtp_path to the dmail directory. The dmail directory is then used to store the manual, the utility executables and almost all parts of DMail, except for the config file dmail.conf.

On Windows NT it will also, by default, create the following directory structure,

```
dmail   ->      dlist
        ->      dwatch
        ->      in
        ->      log
        ->      work
```

and set the various path settings, e.g. work_path, log_path, dwatch_path, etc. to match.

If the complete DMail package is installed on a Unix machine the sendmail application is saved and replaced by a sendmail stub. The DMSetup wizard can do this for you at installation time. DMSetup will also restore the original sendmail application, if requested, when removing other parts of DMail

Once DMSetup has finished the installation it starts the DWatch program which in turn starts each of the three email servers, DPOP, DSMTP and DList. If you are running DMail on Windows NT then it will also automatically start the DMAdmin GUI server management utility.

OS Installation Notes

All Operating Systems:

- Note that you cannot use spaces in path names in the configuration file, e.g. you can NOT have a dsmtp_path setting of,
  dsmtp_path c:\program files\dmail

Windows 95 and 98:

- See the Special notes at, DMail on Windows 95 and 98.

Red Hat Linux 6.2 or above:

- Use the 'libc6' version of the download. You are probably running the version 6 C libraries so you should be sure to use the downloads marked 'linux_libc6'
- Location of Drop Files: On startup on some systems the default system POP server will move all drop files from /var/spool/mail to $HOME and use a filename of 'mbox', e.g. /var/spool/mail/support to ~support/mbox.

  This can make it seem as if all mail has disappeared when you start running DMail.

  We recommend that you move these drop files to /var/spool/mail.

However, you can use the ~ symbol in the drop_path setting to specify that dsmtp uses the user's home directory as the drop path, e.g.
drop_path ~/mbox
will result in mail for user 'support' going to ~support/mbox.

MAC OSX:

- Ensure that you run the installation from a 'Terminal' command line prompt NOT using a GUI Admin tool!
- Note that you cannot use spaces in path names in the configuration file, e.g. you can NOT have a dsmtp_path setting of,
  dsmtp_path /Local/Library/Mail Server/
- See the Startup_Scripts for information on making DMail start up on reboot

---

After Installation - What Next?

Here is a list of things to look at or consider after installation ...
(on this page)

1. Adding Users
2. Your DNS MX Entries
3. Add Relay Restrictions
4. How to send a test email

(on other pages)

- IMAPD Server Installation
- Making DMail start on reboot (Startup Scripts)
- Sendmail Stub

For those wanting to set up a 'Hotmail' type system see,
Step by Step Install

1. **Adding Users**

   Once you have installed DMail, all users on your system are valid DMail users (given that they meet any access restrictions which you set set in dmail.conf). Therefore you can add DMail users in the fashion that you would normally add users to your system. On windows NT, you go to START -> Programs -> Administrative tools -> User Manager, then select "New user" from the user menu. On Unix you can use the newuser <username> command.

   Notes:

- if you have set, authent_domain true, in dmail.conf configuration file, then you should add usernames to the user database as,
  user@domain
  rather than as just 'user'. Where the domain to use is the **first** host_domain line in dmail.conf.
- If you want to use the 'Users' button in DMAdmin, then you have to be using External Authentication which you should set manually in dmail.conf, see External Authentication.

2. **Your DNS MX Entries**

   You should set up DNS MX records that point at your new email server for every domain that you intend to administer.

   Then make sure that you add either a host_domain OR a (virtual) vdomain line in dmail.conf for each of these domains.

   E.g. for two real domains, domain1 and domain2, and two virtual domains, domain3 and domain4, your dmail.conf would have these lines:
   host_domain domain1
   host_domain domain2
   vdomain vdom3 x.x.x.x domain3 /mail/dom3
   vdomain vdom4 x.x.x.x domain4 /mail/dom4

   Basically host_domain settings tell DSMTP what other domain names it should consider as local domains, i.e. they are synonyms for your main domain. E.g. bob@domain1 also has the address bob@domain2.

   Whereas the vdomain lines specifiy local virtual domains which are domains other than your main one which you wish to host on the same machine, e.g. bob@domain1 and bob@domain3 are different users.

   Please see the section,
   Domain Name Resolution (DNS),
   for further information on what you need to set up and how DSMTP uses the DNS server.

3. **Add Some Relaying Restrictions**

   The default behaviour for DSMTP is to run without any relaying restrictions. You should add some restrictions, before your email server becomes a spam gateway for all the bored people of this world.

   We suggest you start with
   forward_from_ip 127.0.0.1
   and
   forward_from_ip x.x.x.*
   where x.x.x is the first three setions of the IP address of your machine.

Note: In version 2.7q and above, DMSetup will automatically add these two lines for you.

This will restrict people from giving your SMTP server mail to be delivered to any other server, i.e. non-local mail, unless they are connecting from your IP address.

Note: As soon as you add one relaying restriction rule, then DSMTP prohibits all relaying unless it is expressly allowed by a relaying rule. So the default setup is that only mail coming from your local IP Address can be relayed.

See the Spam Rules section for information on this.

4. **How to send a test email**

To test your mail system, you should try sending an email to a local user.

To do this you should use a normal email client, e.g. Pegasus Mail, Eudora, Netscape Mail etc., and point it at the new DMail server. Because you probably don't yet have a DNS entry for your domain that points to the DMail server, you should tell your email client to use the IP Address of the DMail server as the 'sending SMTP server or outgoing mail server'.

You should then send an email to the address,
username@domain
where

1. 'domain' is the domain given by the first host_domain setting in the dmail configuration file, dmail.conf (click here to find out where to locate your dmail.conf).
2. and 'username' is a username that you have added to the user database.

You should also try logging in to DPOP to collect the mail message for that user.

Again, point your email client at the IP Address of the DMail server, and provide a username that matches the username part of what you added to the user database, e.g.
bob
with corresponding password.

NOTE: if you have set authent_domain true then you will have added, user@domain to the user database, **but** you login to DPOP with 'user', not 'user@domain'.

If any of these tests fail then please do the following,

1. set,
   log_level debug
in dmail.conf and save the file.
2. reload both DSMTP and DPOP, by issuing the commands,
   tellsmtp reload
   tellpop reload
at a command line.
3. test sending in a message and trying to collect it.
4. send us the resulting log files,
   dsmtp.log

dpop.log
which you will find in the log_path directory as set in dmail.conf
5. send DMail Support your dmail.conf file, typically,
/etc/dmail.conf or \winnt\system32\dmail.conf

Removing the DMail Package: DPOP, DSMTP, DList

To revert to the previous popper DPOP needs to be shut down and inetd.conf again edited and another SIGHUP sent to inet. These steps can be performed for you by the DMSetup wizard.

One other major issue is that messages read but not deleted during the time DPOP is running will no longer be stored in standard drop files, but instead in its own "bin" files. Before reverting to a previous popper the DPOP bin files need to be converted back into standard drop files. This is done by using the **Tellpop drop_all** command.

The recommended safe-removal procedure for DSMTP/DPOP/DList is:

1. tellpop offline
2. tellsmtp tryall
3. tellpop status
   (Repeat until current users = 1)
4. tellsmtp status
   (Repeat until no message domains are listed, i.e. all outgoing messages have been sent)
5. tellpop drop_all
6. dmsetup
   Choose option 3 to remove DPOP / DSMTP / DList
7. Restart original popper and check operation
8. Remove unwanted files from directories identified by 'dmsetup' two steps above.

On Unix based systems, you may have allowed DMSetup to save the sendmail application and replace it with a Netwin sendmail stub, or you may even have done it manually. If so, DMSetup will also restore the original sendmail application, if requested, when removing other parts of DMail

---

Lib C 6 - Linux

Versions 2.4j and upwards have been built for Linux platforms with the Version 6 C libraries, found on Red Hat Linux 5.2 as well as with the older libraries.

Such versions are referred to in the documentation and on the download page, as the libC6 versions, e.g.

dm24j_linux_libc6.tar.Z

LibC6 versions are required if you intend to use shadow passwords and the Yellow Pages system for shared password files.

They also fix the time zone problem - see Time Zone problems on Unix

Sendmail Stub

The DMSetup installation wizard tries to replace the sendmail process with our 'stub'.

Because Sendmail has a command line emailing component not present in DSMTP, we provide a command line email client, which you can use to do any command line emailing. Users who telnet directly to the mail server machine and CGIs commonly use command line emailing.

What should I do if DMSetup fails to install the sendmail stub?

DMSetup tries to replace,
/usr/sbin/sendmail
with our stub also called,
sendmail

The way to tell the two apart is by file size, our stub is only about 40k where as the Sendmail binary is of course a lot bigger.

You will find our stub in the temporary unpack directory of the distribution set, dmtemp/sendmail. You should copy it manually over the location of your sendmail binary file, if DMSetup fails to install it.

You should also check that any symbolic links to the sendmail stub, such as the 'mail' command also now point to this stub.

When run, our sendmail stub logs to a file, sendmail.log, in the dsmtp_path directory. So to check that it is being run you would typically examine the file,
/usr/local/dmail/sendmail.log

Configuration Settings Specific to DSMTP:

This page details only those dmail.conf settings which apply specifically to DSMTP.
For more settings see: Settings Common to DPOP and DSMTP, Settings Specific to DPOP and Settings Specific to DList

Note: After modifying a DSMTP setting you should do a tellsmtp reload command or send a reload command to the DSMTP server using DMAdmin.

Compulsory settings:

(If omitted, DSMTP may function unpredictably, or not at all.)

host_domain	NOW a COMMON setting to both DSMTP and DPOP
dsmtp_path	Location of DSMTP executable (and default for many path settings).
sysadmin	The email address of the system administrator and 'postmaster' for DSMTP.

Often Used Optional settings:

(These settings may be omitted, and all have reasonable defaults)

ban_ip Specifies an IP address that DSMTP should not talk to. ban_mailfrom Specifies patterns for banned mail from lines. ban_rcptto Specifies patterns for banned rcpt to lines.
bomb_dec Mail Bomb; Specifies how much to decrement the entries in the mail-bomb cache by.
bomb_dir Mail Bomb; Specifies which directory mail-bomb messages are to be stored in.
bomb_entries Mail Bomb; Specifies how many entries to store in the mail-bomb detection cache.
bomb_max Mail Bomb; Specifies DSMTP's tolerance for detecting a potential mail-bomb.
dns_host Set specific DNS servers to be used for domain lookups (instead of using system DNS settings).
dns_timeout Specifies how long (in seconds) DSMTP should wait on a DNS lookup.
drop_max Specifies how big DSMTP should let dropfiles get (in kbytes).
log_data Specifies whether or not DSMTP should log the TCPIP data it gets.
max_msgsize Specifies the maximum message size DSMTP may accept(in kbytes).
max_rcpts Use to limit the number of recipients DSMTP should allow per message.
max_retry *This command is obsolete, use* max_retrytime
max_retrytime Specifies the maximum time in hours to continue attempting (once every 2 hours) to

deliver a message.
msg_filter Tells DSMTP to use a message filter (in specified file) on all incoming mail.
quiet Tells DSMTP to direct minimal output to stdout
robot_try Specifies how long DSMTP should try to give input to a robot before giving up (in seconds).
robot_wait Specifies how long DSMTP should wait before killing a robot.
shell_prefix Robot; tells DSMTP to use a prefix for autoresponder exec calls.
smtp_port Allows you to alter to a non-standard value the port that DSMTP will listen for SMTP connections on.
tarpit_start Anti-Spam; Number of Recipients before DSMTP should start responding slowly.
tarpit_except Lists IP addresses which are exceptions to tarpit_start.
tcp_max Specifies the maximum number of TCPIP channels that DSMTP listens on.
tcp_timeout Specifies how long DSMTP should wait (in seconds) for a response before giving up on ALL TCPIP connections (except sendlog connections).
timezone Tells DSMTP to fake its timezone info with this value.
use_forward_files Used to stop DSMTP from checking for users' forward files.

Obscure Optional settings:

(These settings may be omitted, and all have reasonable defaults)

add_footer Appends the given footers file if the envelope from matches the given domains.
alias_file Specifies file where DSMTP is to look for user alias information.
alias_file_domainSpecifies file where DSMTP is to look for domain specific, user alias information.
alt_drop_path Specifies an alternative dropfile path for a particular domain (use only when not using external authentication).
auth_allow Sets various permissions for ESMTP/AUTH success.
auth_hide Switches off the announcing of ESMTP AUTH under various conditions.
bounce_body Specifies whether or not to include the message body in a Deliver Status Notification message (DSN).
bounce_maxlen Specifies (if bounce_body is true) how much of the body (in kbytes) to include in a Deliver Status Notification.
domain_chroot (UNIX) Robots; Enables chroot functionality for domains.
dotstuff_robot Makes DSMTP dotstuff robots for backwards compatibility.
external_processor Activates DSMTP's message processing daemon functionality.
external_viruschecker Tells dsmtp to use an external virus checker on MIME attachments from extract_mime
fallback_address Enables a 'catchall' address for a specific domain.
forward Creates a mail re-direction rule to be checked against all incoming mail.
forward_cc Creates a carbon-copy mail re-direction rule to be checked against all incoming mail.
forward_from Anti-SPAM; specifies a domain that is exempt from relaying restrictions.
forward_from_ip Anti-SPAM; specifies an IP address that is exempt from relaying restrictions.
forward_user Anti-SPAM; turns on an anti-relaying system that allows relaying only after a recent

DPOP login.

forward_window Anti-SPAM; sets time (in seconds) in which relaying is allowed after POP login (i.e. expiry time of forward_user records).

fromip_max Anti-SPAM; sets a limit on message throughput per hour from any IP number.

fromip_nolimit Anti-SPAM; specifies IP addresses which are exempt from the fromip_max setting.

gateway Bypass DNS lookups by specifying domain-IP_address pairs for DSMTP to use.

lock_id (UNIX); sets a file locking id for multi-server systems.

log_days Tells DSMTP how long (in days) to keep summary log files.

maps_action Invokes certain actions should the connecting server be registered with MAPS RBL.

max_loglen Specifies the maximum size (in kbytes) of DSMTP's log files.

max_others Specifies the maximum number of recipients recorded for each message in summary files.

max_queue Sets the number of rcpt lines from queue files that DMSTP queues up in memory.

max_rcvd Anti-SPAM; Specifies the maximum number of received lines allowed in an incoming message.

max_send Sets maximum outgoing channels, which limits maximum simultaneous outgoing messages.

min_space Specifies the minimum disk space (in Mbytes) DSMTP needs to operate.

no_autohost De-activate the auto adding of hosts to DSMTP's internal host_domain list.

no_dotforward (UNIX) tells DSMTP not to look for .forward files but .fwd files instead.

oneline_stats Makes DSMTP use a single line per message in new format for its sent message log - for tellpop statistics command.

orbs_action Invokes certain actions should the connecting server be registered with ORBS.

hide_rcvd_ip Makes DSMTP hide the specified IP address in the 'received' message header.

ras_domain RAS Dialup; This setting is NOT necessary in almost all cases. It specifies the domain on which authentication is to occur (NT only).

reject_no_reverse Tells DSMTP if it should reject messages where reverse DNS on connecting IP address fails (requires reverse_lookup to be set).

relay_to Permits unconditional relaying to particular domains.

remind_timeout Specifies the minimum time (in seconds) between critical error emails.

rotated_logs Sets the number of old log files dsmtp keeps.

show_8bitmime; Makes DSMTP revert to pre version 2.5c default of advertising 8 bit MIME capability (not recommended).

show_ehlo; Switches on the announcing of various ESMTP extenstions.

smtp_welcome; A template for the welcome line. Accepts \\r\\n delimited lines, and $DATE, $HOST and $QFILES macros.

spool_dir; Activates a mail spooling directory for incoming mail. DSMTP will deliver any mail found in this directory..

unix_case (UNIX)Tells DSMTP to use strictly case sensitive user lookups.

user_quota Enables a per-user mailbox quota system and optionally sets the default quota(true/false/kbytes).

vdomain_passwd Common setting to DPOP and DSMTP

virtual_user_post Adds a sendmail style virtual user table, actioned immediately before lookup.

virtual_user_pre Adds a sendmail style virtual user table, actioned before any other local-user rules

apply.
warn_user Specifies when (after x hours) DSMTP is to warn the sender of a delayed delivery.

Multiple entry settings: These settings may occur more than once(repeated from lists above).

add_footer Appends the given footers file if the envelope from matches the given domains.
alias_file_domain specifies where DSMTP is to look for domain specific user alias information
alt_drop_path specifies an alternative dropfile path for a particular domain
ban_ip Specifies an IP address that DSMTP should not talk to.
host_domain adds a domain name to the list of domains to be recognized as being local
dns_host Set specific DNS servers to be used for domain lookups (instead of using system DNS settings).
fallback_address enables a catchall address for a domain
fromip_nolimit enables exceptions to fromip_max
forward creates a forwarding rule for all incoming mail
forward_cc creates a carbon-copy forwarding rule
forward_from establishes relaying restrictions
forward_from_ip establishes relaying restrictions
gateway creates a gateway type rule for all incoming mail
relay_to permits unconditional relaying to particular domains
vdomain_passwd common setting to DPOP and DSMTP

Detailed Descriptions of DSMTP Configuration Settings:

---

add_footer <filename> <domain>[,<domain>,...]

This setting specifies a footer file to be added onto the end of all messages sent out by DSMTP, from users on any of the specified domains.

The <filename> can be specified with a path, or without. If a path is not given then dsmtp looks in the directory specified by the dsmtp_path setting.

(NB: DON'T specify a path in version 2.8b, i.e. put the footer file in your dsmtp_path directory.)

One or many domains can be specified. Use a comma separated list (no spaces) to specify a list of domains to which the setting should apply. In addition you can use one of the keywords to save typing,
host - means that dsmtp should add the full list of host_domain settings to the list
virtual - means that dsmtp should add the full list of vdomain domains to the list

default - means that this should be applied to any domain that does not have a specific add_footer setting.

Example1:
add_footer footer.txt domainx.com
would add the footer file, dsmtp_path/footer.txt onto any outgoing messages sent by users on the domain, domainx.com

Example2:
add_footer footer.txt default
would add the footer file, dsmtp_path/footer.txt onto any outgoing messages sent by users on any domain (unless their domain had a specific footer of its own as set by another add_footer setting).

Example3:
add_footer footer.txt host,domainx.com
would add the footer file, dsmtp_path/footer.txt onto any outgoing messages sent by users on any host_domain as well as those on the domain, domainx.com (presumably a virtual domain).

---

alias_file <filename>

This setting tells DSMTP which file contains the user alias information. The **<filename>** parameter must include the full pathname and the filename of the original alias file. DSMTP makes a copy of this file in aliases.dml, in its working directory to use when doing user alias lookups. Alias files are textfiles containing lines of the following format:
user: destination
e.g.
username: user@domain
username: /files/path/dropfile
username: |/usr/bin/autoresponder

Lines starting with a hash are treated as comment. Destinations that have multiple words should be within quotes. e.g.
username: "|/usr/bin/autoresponder arg1 arg2"

NOTE: All aliases found for a user by DSMTP are applied.

You can include another alias file using the :include: in the destination, e.g. to include the alias file /etc/aliases2 you could use the following line,
username: :include:/etc/aliases2

As the examples above show you can use an alias in the same way as you can use a forward setting to make DSMTP write incoming messages directly to file or to pass them to a robot with the pipe symbol, | .

Note: The alias must be entered without a domain, but the destination for the alias should contain a

domain - otherwise DSMTP will assume the domain is the domain given in the first host_domain setting. If you do not wish to specify the destination domain and you don't like the host_domain default then you should put the alias in a domain specific alias file, identified in dmail.conf with the alias_file_domain setting.

You do not need to enter virtual domain prefixes in the alias file. E.g. with a vdomain prefix of dom1_ the following line is incorrect,
dom1_bob@domain1.com: dom1_fred@domain1.com
it should be,
bob@domain1.com: fred@domain1.com

Aliases only apply to valid local users, so DSMTP will check that an incoming message is for a valid local domain before it checks for an alias, whereas a forward rule can apply to local or non-local users, i.e. any domain.

For more information see the Forwarding and Aliasing section.

Example:

alias_file /usr/aliases

---

alias_file_domain <domain> <filename>

This setting works exactly the same way as alias_file, except that the alias file **<filename>** is only applied to domains matching **<domain>** which may include wildcards.

This allows you to set up alias files for specific domains and also to create global aliases.

To create global aliases you can add the setting,
alias_file_domain * global_aliases and then within the file, global_aliases, you can add aliases that you want to apply on ALL domains.

NOTE: All aliases found for a user by DSMTP are applied.

NOTE: You do not need to enter virtual domain prefixes in the alias file. E.g. with a vdomain prefix of dom1_ the following line is incorrect,
dom1_bob: dom1_fred@domain1.com
it should be,
bob: fred@domain1.com
or even just,
bob: fred

If a domain is not specified on the destination, e.g.,
bob: john
then DSMTP will assume the destination user is on the same domain.

In version 2.7m and above you can use the syntax,
alias_file_domain domain filename suffix
where suffix is appended to any alias destinations with only subdomain names, e.g.

bob: fred@machine1
becomes a legal setting if the suffix was set to
.domain.com
, i.e. fred@machine1.domain.com.
NB: dsmtp only adds the suffix if there is no dot in the destination domain.

For more information see the Forwarding and Aliasing section.

Example:

alias_file_domain /usr/aliases

---

alt_drop_path <domain> <pathname>

This setting specifies an alternative path to use for the dropfiles of messages whose destination is **<domain>** The **<domain>** parameter may contain the wildcard character '*'. The path named by the pathname parameter must exist. This setting will be ignored if authent_method is set to external. This setting can be used any number of times.

Note: This setting will be ignored unless the **<domain>** parameter is specified as being local by using the host_domain setting. The last applicable setting will be used.

Example:

alt_drop_path *.spammers.com /etc/trash/

---

auth_allow [relay],...

This setting specifies various permissions or actions that are allowed for any user that has authenticated to DSMTP using the SMTP AUTH command.

This setting is a comma separated multiple value setting which can take any of the following key words:

Keyword	Allows the authenticated channel or user to ...
relay	relay mail to non-local domains.

If 'relay' is set then DSMTP will automatically advertise,
AUTH PLAIN LOGIN
in response to the EHLO command. You can control this with the setting, show_ehlo.

More about SMTP AUTH ...
SMTP AUTH allows users to turn SMTP AUTH on in their email client. SMTP AUTH means that the email client will provide the username and password (same as on DPOP server) to authenticate on your DSMTP server when connecting to send out mail.

NB: adding this setting will mean that some email clients like Netscape Mail force the users to turn on SMTP AUTH. Generally this is not a problem as Netscape Mail instructs them on how to do it, but it may be confusing to some users.

If using the forward_user system as well then you should probably set the setting,
hide_auth recentpop
so that unnecessary auth lookups are not done.

We also have a new proxy widget called SmtpAuth (currently only in windows beta form) which takes a username and password to authenticate to an SMTP server with.

So users with an email client that does not support the SMTP AUTH command can run this on their machine and point their client at it instead of directly at your smtp server. It then authetnicates to your server before sending on any mail feed to it.

If it is a whole domain coming through another trusted server then they could use the SmtpAuth proxy and feed all their outgoing mail through it. As we only have SmtpAuth on NT their server would have to be running on NT. If their server is DSMTP then we plan to add a setting so that DSMTP 'auths' all outgoing connections to a given ip address. So contact DMail Support if you need that setting.

Example:

auth_allow relay
Allows any user that authenticates successfully to relay messages to non-local domains bypassing any other relaying settings like forward_from_ip.

---

auth_hide <ipnumber>,<ipnumber>,...["recentpop"]

This setting switches off the announcing of ESMTP AUTH under various conditions, so that AUTH lookups are not done when not necessary.

You can use <ipnumber> to specify ip addresses. It can be,
x.x.x.x
or
x.x.x.* (a wildcard)
or
x.x.x.y-z (a range)

This setting is a comma separated multiple value setting which can also take the following key words:

Keyword	Meaning ...
recentpop	add all ip addresses in the forward_user POP before SMTP system to the list

The 'recentpop' keyword, makes DSMTP check the ip address of a connection against its list of ip addresses which have recently successfully logged in to the DPOP server. This setting is only of use if you have set, forward_user to be true.

Example:

auth_hide 127.0.0.1,1.2.3.4,recentpop
This will stop DSMTP from advertising SMTP AUTH to those users connecting from local IP addresses, 127.0.0.1 and 1.2.3.4, as well as anyone connecting from an ip address that has recently

checked for mail on the DPOP server.

---

ban_ip <IPaddress>

This setting specifies an IP address that DSMTP should not talk to. If something attempts to connect to DSMTP from this IP address, DSMTP serves them with this line:

551 You have no permission to talk, Goodbye

at the end of which the connection is dropped by DSMTP.

Example:

To stop anyone connecting from the IP address 1.2.3.4, use

ban_ip 1.2.3.4

---

ban_mailfrom <string>

Specifies patterns for banned mail from lines. If set, dsmtp will scan all MAIL FROM: lines in the SMTP envelope to see if the sender's email address contains the given string. If it does then DSMTP will bounce the message.

NB: DSMTP does not check the message headers for the specified string.

This setting can be given multiple times in dmail.conf.

Example:

ban_mailfrom bob.com

will make dsmtp bounce all messages which are sent by a user who's address contains the string 'bob.com'.

---

ban_rcptto <string>

Specifies patterns for banned rcpt to lines.If set, dsmtp will scan all RCPT TO: lines in the SMTP envelope to see if any of the message recipients' email addresses contain the given string . If they do then DSMTP will bounce the message to only that specific recipient.

NB: where a message is to multiple recipients DSMTP will still deliver the message to any other recipients that do not match the given string.

NB: DSMTP does not check the message headers for the specified string.

This setting can be given multiple times in dmail.conf.

Example:

ban_rcptto bob.com

will stop dsmtp from delivering mail to any destination address (given in the envelope RCPT TO: line)

that contains the string 'bob.com'.

---

bomb_dec <number>

This setting is an extension to the bomb_max setting. It is used for catching particularly devious mail-bombs which arrive over an extended period. The setting tells DSMTP how thoroughly it should clean up its sender-recipient cache. Once an hour, DSMTP goes through the cache and subtracts **<number>** from each counter (the entry will not be removed, even if the counter becomes 0). Under most circumstances, the **<number>** parameter should be set to be the same as the bomb_max setting. This assumes that any mail-bomb will take the form of a surge of messages in a short time. Some particularly devious sorts might send one message every five minutes for a day, which would still be a mail-bomb. If this is suspected to be happening, setting **<number>** to be much lower would trap the incident. Extreme care must be taken with this setting however, as DSMTP **will not deliver** messages it thinks are mail-bombs, and it will only keep the last 10 it has received. Basically, if DSMTP receives bomb_max messages in (bomb_max/bomb_dec) hours, it will trigger a mail-bomb alert. The default value is 50.

Example:

With bomb_max 50 if we want to define 50 messages in 5 hours as a bomb we set bomb_dec to 10 (as 5 = 50/10): bomb_dec 10

---

bomb_dir <pathname>

This setting tells DSMTP where to put rejected mail-bomb messages. A maximum of bomb_max messages will be placed here, the last one being the most recent. The default value is work_path

Example:

bomb_dir /dev/null

---

bomb_entries <number>

This setting tells DSMTP how many sender-recipient pairs to cache for mail-bomb detection. Every time DSMTP receives a message to be delivered locally, it stores the sender-recipient pair. If the pair is already present, it increments the associated counter. If there are **<number>** entries already in the cache, DSMTP finds one with a low count and deletes it to make room for the new one. The larger this setting is, the more likely DSMTP is to detect a mail bomb. Performance degradation may occur if it is made too large. The default value is 2000.

Example:

bomb_entries 500

---

bomb_entries <number>

This setting tells DSMTP how many sender-recipient pairs to cache for mail-bomb detection. Every time DSMTP receives a message to be delivered locally, it stores the sender-recipient pair. If the pair is

already present, it increments the associated counter. If there are **<number>** entries already in the cache, DSMTP finds one with a low count and deletes it to make room for the new one. The larger this setting is, the more likely DSMTP is to detect a mail bomb. Performance degradation may occur if it is made too large. The default value is 2000.

Example:

bomb_entries 500

---

bomb_max <number>

This setting defines what DSMTP is to call a mail-bomb. If more than **<number>** messages are received for the same user, from the same source in a one hour period DSMTP will alert the system administrator and intercept any further messages with the same sender-recipient pair. These will be placed in the directory specified by the bomb_dir setting. To disable this feature, set **<number>** to something big, like 10000. The default value is 50.

Example:

bomb_max 20

---

bounce_body <switch>

This setting tells DSMTP whether or not it should include the body of a message in its Delivery Status Notification (DSNs can result from a failed delivery (bounce) or a successful one, if requested by the sender as set out in the Extended SMTP RFC). DSN requests are added by the sender's email client to the MAIL FROM: line of the ESMTP envelope, e.g.
MAIL FROM: <bob@netwinsite.com> RET=FULL
would request the full body to be returned with the DSN, and RET=HDRS requests that only the headers be sent if there is a DSN.

This setting can take true, false (= never) or always as an argument, with the following meanings:

true - **if requested** by the sender, send the message body with the notification message.
false - **never** send the message body with the notification message, even if it is requested by the sender.
always - **always** send the message body with the notification message, even if the sender only requests that the headers be sent.

The default is false, which means only the headers will be returned to the sender on a bounce or DSN.

Example:

bounce_body false
will mean that the body of a message is never bounced even if the user requests it as part of the ESTMP DSN.

---

bounce_maxlen <number>

This setting specifies when to truncate the body of a bounced message. If the message body exceeds

<number> kb, DSMTP will truncate it. The default is 20.

Example:

bounce_maxlen 5

---

dns_host <ip number>

NOTE: By default DSMTP will use any DNS servers you have set up your operating system to use (this includes any domains listed in your hosts file (commonly /etc/hosts or \winnt\system32\drivers\etc\hosts). So you probably don't need to use this setting.

This setting is provided so that you can specify a list of DNS servers that DSMTP should use to lookup domains (resolve them to an IP address) INSTEAD of any DNS servers setup for use in the operating system.

For outgoing mail DSMTP will attempt to connect to the machines specified in email messages by using this setting for all DNS lookups, **except** for any that match those given by the gateway setting.

This setting can be used any number of times or take a comma separated list of IP addresses. If multiple hosts are given, DSMTP will go through them all until it gets a valid answer.

The default is to use the system DNS setup, i.e. no dns_host setting.

Example:

dns_host 127.0.0.1

---

dns_timeout <time>

This setting tells DSMTP how long to wait (in seconds) before giving up on a DNS lookup. If **<time>** is set too low, DSMTP may record a lookup failure where none occurred. Messages addressed to a domain whose lookup failed are placed in the retry queue. Repeated failures will cause DSMTP to bounce the message. If this appears to be happening too often, try adding another DNS host to dmail.conf, or increasing the **<time>** parameter. The default value is 10 seconds.

Example:

dns_timeout 20

---

domain_chroot <domain> <path> (Unix based platforms only)

It is possible to configure DSMTP to make use of user-maintained alias files (using alias_file_domain for instance) and forward (.fwd) files. These can include references to files and/or autoresponders. Particularly in the case of virtual domains, it may be necessary to chroot to a particular directory before performing the desired operation. **domain_chroot** does this. The **<domain>** parameter may contain wildcards, DSMTP does **not** check that **<path>** exists.

Example:

vdomain prefix 1.2.3.4 vdomains.r.us.com /usr/local/vdomsrus/var/spool/
domain_chroot vdomains.r.us.com /usr/local/vdomsrus/

If a domain has a domain_chroot setting then references to robots and different drop files in forwarding rules and the like, can take relative paths. E.g. in the example above, the domain_chroot for vdomains.r.us.com would mean that a forward rule for a user in that domain to a drespond robot would have to call the robot from a relative path, with its root based at the chrooted domain, i.e. /usr/local/vdomsrus, e.g.
forward bob@vdomains.r.us.com "|/drespond /message.txt"
where /drespond means run the drespond program from the root of the chrooted directory, so the file being run is,
/usr/local/vdomsrus/drespond
and the message file is similarly located in the same directory.

NB: domain_chroot is intended for Unix based platforms only.

---

dotstuff_robot <true/false>

In versions 2.5d and 2.4k DSMTP's default behaviour was changed on Unix platforms so that it no longer does dotstuffing on messages going to robots. On Windows platforms the dotstuffing of messages to robots still occurs. This setting is provided for backwards compatibility only. If set true then DSMTP will carry out dot stuffing on messages that it writes to robots.

Example:

dot_stuff true

---

drop_max <number>

This command tells DSMTP how big to let dropfiles become. If a dropfile is bigger than **<number>** kb, DSMTP will not deliver the message (it will alert the sender to the problem, but not the recipient). This setting should be used in conjunction with user_quota because when a user checks their mail, DPOP clears the dropfile and sorts the messages within it into its own 'bin' files. So a user can have an empty dropfile but still use up a lot of space in message storage on your email server.

Example:

drop_max 5120

---

dsmtp_path <pathname>

This is the DSMTP installation directory. It will contain the DSMTP executable, help files and utility executables. The work_path and log_path default to here. The DMSetup installation wizard will by default create a directory called 'dmail' and point all of the server's home directories at it, i.e. dpop_path, dsmtp_path and dlist_path are all made to point to the directory called dmail by default.

This setting applies indirectly to DSMTP,DPOP and DList. It is global across all domains and by default cannot be changed by domain administrators.

Example:

dsmtp_path c:\mail\dmail\

---

external_processor <true/false>

This activates dsmtp's message processing daemon functionality. This allows you to run a process on all incoming mail before dsmtp delivers it to local or non-local users.

When this setting is set to true, DSMTP writes a file called m_x.dmn in the work_path directory when it receives the closing dot of an incoming message. It will write the incoming message to a temporary file, m_x.tmp.

NB: the value x will be a unique number starting at 0, e.g. 0 or 1 or 2 or, 99999 etc.

It is then up to the external processor (daemon) to detect this file, and process m_x.tmp, i.e. a file of the same name but with the ending '.tmp'.

NB: it is important that the external processor does not touch the .tmp file until the .dmn file appears, as dsmtp may still be writing to it.

The processor can make whatever modifications to the message file m_x.tmp that it sees fit, including removing the file altogther. Common things are removing objectionable content, infected attachments, adding/removing footers, altering/removing headers, etc. The only priviso is that if it leaves the file it must leave a valid message otherwise DSMTP may not be able to deliver the message.

DSMTP waits for a file, m_x.rdy to be created by the external processor. When that file arrives, dsmtp processes the message according to what it finds in the .rdy file.

1. If the .rdy file is empty then dsmtp will deliver the message it finds in the m_x.tmp file.

2. If the .rdy file is empty and the .tmp file has disappeared then DSMTP will bounce the message and give a permanent error response code (i.e. 500 level).

3. If the .rdy file has a first line with an SMTP-style reply line, dsmtp will use that response code and message for it's SMTP response to the message data and not deliver the message, i.e. dsmtp will bounce the message with that response code. A negative SMTP-style reply line starts with a '4' or '5', i.e. >=400

It is possible to run multiple daemons. The first daemon can write a m_x.2 file to trigger a second daemon, so and on, until the final daemon writes the m_x.rdy file.

NB: DSMTP freezes the incoming message channel until it sees the .rdy file and knows how to respond. So the external_processor has up to tcp_timeout seconds to process the message and create the .rdy file. It it does not DSMTP will drop the TCPIP connection and tidy up the external_processor files. In practice it probably has a much shorter time to deal with the message, because email clients generally have a much shorter timeout for sending the message than the tcp_timeout value. Email clients typically have a timeout value of 15-30 seconds.

NB: be careful not to set this setting true if there is nothing there to process the files, - otherwise the mail will never be delivered.

This setting was added in version 2.8i

Example:

external_processor true
causes DSMTP to write incoming messages to a .tmp file in the work path, and create a .dmn file when it has finished. The system administrator's daemon then detects the .dmn file and processes the .tmp file as it wishes. When it is finished it writes a .rdy file. The .rdy file contains a response code if the message is to be rejected or the .tmp file has been deleted by the daemon.

---

external_viruschecker <command_line>

Tells dsmtp to use an external virus checker on MIME attachments from extract_mime.

For every attachment extracted with mime_extract, DSMTP spawns this command line and replaces the macro $FILE$ with the full filename of the extracted attachment file.

NB: DSMTP only spawns the external virus checker on MIME attachments that it checks. It does not run the virus checker on EVERY message.

DSMTP **expects** the virus checker to delete infected files, and does **not** examine the output of the virus checker itself.

If any of the attached files have disappeared, DSMTP will reject the message and send a 'infected attachment' warning to the sender.

Example:

external_viruschecker c:\virus\viruschecker.exe -delete -quiet $FILE$
Tells dsmtp to use an external virus checker on MIME attachments from extract_mime
{M_EXTERNVIRUS, "external_viruschecker", NULL, NULL, 0, F_DSMTP, "V:2.8n !: Tells dsmtp to use an external virus checker on MIME attachments from extract_mime"}, {

---

fallback_address <domain> <address>

This setting sets up a fallback address for a given domain. If a message is sent to a user at the given domain, but that user doesn't exist (or no longer exists), then the message will be sent to the given address.

The FAQ, I want all domain1 email which does not go to a specific user . . . is a common use for this setting.

Example:

fallback_address my.company.com boss@my.company.com

---

fromip_max <number>

This setting activates a restriction on the number of messages an IP number can send through DSMTP

per hour. The default value is 10000.

Example:

fromip_max 50

---

fromip_nolimit <ip number>

This setting allows exceptions to fromip_max for local/trusted IP numbers. Wildcard characters are permitted.

Example:

fromip_nolimit 127.0.0.1

---

forward <wildcard> <address>

Any incoming message whose next destination is of the form user@domain, and matches **<wildcard>** will be sent to **<address>** instead, so long as **<address>** is valid. The **<address>** parameter must be of the form user@domain (i.e. it must be a valid final destination). The comparisons are done upon receiving the RCPT TO: SMTP line, so internally generated messages (such as those generated by the forward or alias settings) will **not** be checked. Any number of these settings may be used. All applicable forward settings will be applied. The '*' wildcard character may be used in the **<wildcard>** parameter. **The message will not be delivered to the original recipient, so if you want the original recipient to receive the message use, forward_cc.**

Note: There is a fallback_address setting that can pick up addresses not covered by a forward rule, see the faq I want all domain1 email which does not go to a specific user to . . .

Forward rules can apply to any users of any domain, not just local users as they must be in an alias file.

New in version 2.8m and above, you can use forward settings for domain rewriting, i.e.,
forward *@domain1 %1@domain2
which would result in mail arriving for 'user1@domain1' being sent to 'user1@domain2', i.e. the domain part of the address is changed to the new domain name.

For more information see the Forwarding and Aliasing section.

Example:

forward bgates@windows.com jreno@doj.gov

will send a copy of all Bill's mail to Janet

---

forward_cc <wildcard> <address>

This setting is matched against all recipients of a message. It causes any message addressed to recipient matching 'wildcard' to be copied to 'address'. This implies it is also delivered to whoever it would have gone to in the first place, including not delivering the message to any address other than the copy addres

or hitting a fallback address.

Note: DSMTP will not deliver the message to the original recipient if a forward rule exists for the same email address, i.e. DSMTP will treat two rules,
forward_cc bob@dom1 bob@dom2
forward bob@dom1 george@dom2
as two forward rules,
forward bob@dom1 bob@dom2
forward bob@dom1 george@dom2

Note: Also the original recipient may not get the message if there is a forward file for that user.

For more information see the Forwarding and Aliasing section.

Example:

forward_cc sales@bob.com bofh@bob.com

---

forward_from <wildcard>

This setting is used to establish an exception to relaying restrictions. If found, DSMTP will allow the relaying of any messages where the domain in the MAIL FROM line of the envelope matches **<wildcard>**
NB: This is not a very strong relaying restriction. The message 'envelope' refers to the information passed as part of the SMTP protocol, so the MAIL FROM line is the line given to DSMTP by the email client (or another SMTP server) and looks like this:
MAIL FROM:
As such, this can be quite easily falsified. See Spam Rules for more details.

Notes:

1. You can add as many of these settings as you need
2. This setting can only be used in versions 2.4h and above.
3. Adding any relaying restrictions, e.g. a forward_from line, automatically turns on relaying restrictions, as the default behaviour is to allow all relaying.
4. You need to make sure that every client and server that connects to DSMTP to send messages OUT has some sort of relaying exception like this forward_from setting (forward_from_ip being the best).
5. In version 2.7 and above, DMSetup will automatically add these two lines for you,
   forward_from_ip 127.0.0.1
   and
   forward_from_ip x.x.x.*
   where x.x.x is the first three sections of the IP address of your machine. This stops your server from being an Open Relay when you have just installed it.
6. In version 2.5f and above this setting is case insensitive, so,
   forward_from domainx.com
   covers, DOMAINX.com Domainx.com etc.

7. A forward rule (e.g. a forward setting or an alias) bypasses any relaying restrictions

Example: To allow relaying ONLY from domains "below" local.domain, say bob.local.domain put this line in dmail.conf
forward_from *.local.domain

to allow relaying from local.domain AS WELL AS domains below it, use this line instead,
forward_from *local.domain

because 'wildcard' is a simple string search.

---

forward_from_ip <wildcard>

This setting is used to establish exceptions to relaying restrictions. If this command is present, DSMTP will not accept mail for non-local addresses unless the sender's ip number matches the **<wildcard>** parameter. If the forward_from_ip rule specifies the sender, then they are allowed to relay messages, i.e. DSMTP is happy to try and send mail to users at other machines for them.

Notes:

1. You can add as many of these settings as you need
2. This command overrides any forward_from commands, e.g. if the sender's IP address matches this setting they can relay no matter what domain they say they are from.
3. Adding any relaying restrictions, e.g. a forward_from_ip line, automatically turns on relaying restrictions, as the default behaviour is to allow all relaying.
4. In version 2.7o and above, DMSetup will automatically add these two lines for you,
   forward_from_ip 127.0.0.1
   and
   forward_from_ip x.x.x.*
   where x.x.x is the first three sections of the IP address of your machine. This stops your server from being an Open Relay when you have just installed it.
5. If you add any forward_from_ip settings you will probably want to add entries for your server's IP address and for 127.0.0.1, so that anything that generates messages on your server can send messages to non-local users, e.g. DList or an autoresponder robot.
6. You need to make sure that every client and server that connects to DSMTP to send messages OUT has some sort of relaying exception like this forward_from setting (forward_from_ip being the best).
7. A forward rule (e.g. a forward setting or an alias) bypasses any relaying restrictions

Example:
To allow relaying only for connections coming from IP addresses starting 1.2.3. add the following rule,
forward_from_ip 1.2.3.*

also to allow messages to be sent to users at other machines from your server, add
forward_from_ip 127.0.0.1
forward_from_ip IPaddress.my.server

---

forward_user <switch>

This setting activates relay permissions for recent DPOP users. If switch is set to true, DPOP will keep a record of all recent logins ('recent' is defined by forward_window . If a message's IP number and FROM envelope entry matches an entry in that record, relaying will bepermitted for that message. This allows users to read and send their mail from anywhere, while maintaing a non-relaying server.

Example:

forward_user true

---

forward_window <number>

This setting tells DPOP how long to let records for forward_user records to linger for in seconds. The default is 120 seconds, or 2 minutes.

Example:

forward_window 240 (sets the window to 4 minutes)

---

gateway <domain> <ip> [return domain]

This setting functions essentially as an instant DNS lookup. If the next destination of a message matches **<domain>** and is not local, DSMTP will not do a DNS lookup on the destination. Instead, it will use the **<ip>** parameter (which must be of the form w.x.y.z).

This can be used to handle fake domains, or force the message to take a particular route. If, for instance, the host machine doesn't have access to the outside world, it can send all non-local messages on to a machine that does. (The **<ip>** parameter must not refer to the host machine. The setting host_domain must be used instead.)

The optional **[return]** parameter, if present, will be added to the messages reverse-path **instead of** the host machine's name. Any bounce messages will then be sent via **[return]** This may be used when the machine pointed to by **<ip>** knows the host machine by some other name (that being what goes in the **[return]** parameter) instead of the host machine's actual name (see host_domain for information on how to set this).

The **last** applicable gateway setting will be used. The **<domain>** parameter may include the wildcard character '*'. Any number of these settings may be used.

In version 2.5g and above the **<ip>** parameter can actually be a domain instead of the ip address. If DSMTP finds a domain destination then it does a DNS lookup on this domain instead of the original domain. The domain may NOT refer a local domain and if it refers to another gateway setting then that setting will be ignored.

Examples:

gateway fake.name 1.2.3.4
will cause any mail arriving for the domain fake.name, to be sent on to the machine with IP address,

1.2.3.4

gateway * 1.2.3.4
will cause DSMTP to gateway ALL mail onto the server at the IP address 1.2.3.4

gateway domain2.com 1.2.3.4 outside.com
will cause DSMTP to gateway mail for the domain domain2.com onto the server at the IP address 1.2.3.4, but before it sends the message it will add outside.com to the reverse path instead of its domain.

---

lock_id <id>

When running multiple DPOP or DSMTP servers that access the same mail spool, it is necessary to give each server a unique id for file-locking purposes. **lock_id** does this. **<id>** can be any string, but for clarity, it should be a number to identify which server is using it.

Example:

lock_id 02

---

log_data <switch>

Normally, DSMTP doesn't log every line sent or received via its TCP channels, as it can add significant bulk to the logfiles. However, full logging can be turned on by setting **<switch>** to 'true'.

Example:

log_data true

---

log_days <number>

This setting tells DSMTP how many days worth of activity summary logfiles to keep. These files are called dmxxyy.sta and dmxxyy.log, where xx is the day and yy the month that the logfile is for. The sta files contain message statistics (number received, sent, local, failed etc), the log file contains lists of recipients and senders for all messages that day. They are retained for accounting and security reasons, but can occupy quite a lot of space if left alone. Any files older than **<number>** days are deleted by DSMTP. The default is 14 days.

Example:

log_days 8

---

max_loglen <number>

DSMTPs logfile can get very large. Once they exceed **<number>** kb, DSMTP rotates out the old one and starts a new one. The default value is 1024.

Example:

max_loglen 2048

---

max_msgsize <number>

This setting tells DSMTP the maximum size a message is allowed to be for DSMTP to accept it. This applies to all messages, not just those being delivered locally. If an incoming message exceeds **<number>** kb, any further data will be discarded. Once the message has finished, DSMTP will notify both the sender and the recipient that the message could not be delivered, with the first 20 lines from the body of the message appended. The default is 2048 (that's 2 megabytes).

Example:

max_msgsize 10240

---

max_others <number>

This setting tells DSMTP the maximum number of recipients per message to log in the daily summary logfile dmxxyy.log(see log_days . If a significant portion of the messages going through DSMTP are mailing list messages with several dozen recipients, the summary logfile can become very large, as each recipient will be recorded for each message. max_others caps the number of **recorded** recipients to **<number>** The default is 20.

Example:

max_others 100

---

max_rcpts <number>

One way to identify spam is that they often have a **very** large number of recipients per message. To trap this, DSMTP can limit the number of recipients an individual message can have. Any further RCPT lines will be refused. Be careful when using **max_rcpts** as legitimate mailing lists can often have large recipient lists. By default, there is no limit.

Example:

max_rcpts 200

---

max_queue <number>

Sets the number of recipient lines from queue files that DMSTP queues up in memory.

Default is 1000 - perfect for 99% of systems. This setting is available in version 2.7k onwards.

NB: Do not play idly with this setting :-)
This setting is provided for tweaking of delivery performance. For example if a server has to deal with a number of large mailing lists you might want to increase it - carefully monitor any changes that you make to it.

All messages that arrive at DSMTP's door get put in the work_path directory as queue file (in pairs - q_x.itm being the message and q_x.idx being an index of the envelope etc.). The way that DSMTP deals with these messages is complex. This setting allows you to change how many of the messages from the

queue are stored in memory (the actual message is not stored in the queue, just information on it). DSMTP does things like scanning the internal queue reordering it, moving messages on and off it etc, so a bigger queue does not necessarily lead to improved efficiency.

See also the queue file section, under Disk Use and Files.

In version 2.7l we have made a considerable change to the efficiency of DSMTP's queue. So if you are moving to this version or higher and you are not running the default for this setting you should re-check its value is correct for your system.

Example:

max_queue 1000
will make DSMTP internally queue up to 1000 of the rcpt lines stored in queue files

---

max_rcvd <number>

This setting sets the maximum number of Received: lines that can be in an incoming message's headers before DSMTP bounces the message. Received headers are added by SMTP servers when they receive a message, for this reason it is used to check that the message has not ended up in a loop. A loop example is where you have two SMTP servers forwarding the same message back and forth to each other.

By default DSMTP rejects (bounces) messages with 15 Received: header lines. These days SMTP servers can generally talk directly to one another, so one or two received lines is normal.

This setting and its default were added in version 2.4e.

Example:

max_rcvd 25
will make DSMTP accept messages with up to 25 Received headers (given it does not reject the message for some other reason).

---

max_retry <number>

This command is obsolete. Use max_retrytime instead.

---

max_retrytime <number>

This setting tells DSMTP how long, in hours, it should continue to attempt to send a message. If the message cannot be delivered after this time, DSMTP will give up. The default is 48 hours.

DSMTP re-tries to send all queued messages every two hours.

Example:

max_retrytime 24
will result in DSMTP attempting to send each message until it is successfully sent or it has unsuccessfully tried 12 times (each attempt being 2 hours apart).

---

max_send <number>

This setting sets maximum number of outgoing TCPIP channels, which limits maximum simultaneous outgoing messages.

The default of 10 is normally adequate. The maximum value is 50.

NOTE: More outgoing channels does not necessarily lead to better sending performance. This setting is provided to allow you to improve performance for your setup, e.g. relays and servers with heavy mailing list loads might require adjustment of this setting. We suggest that you only increment this by 10 channels at a time and carefully moniter any effects.

Use tellsmtp showsend to see a snapshot of the number of outgoing channels in use. Tellsmtp showchans shows you all channels in use, those in state 2 are sends. (those with the socket of -1 are no longer in use).

Example:

max_send 25
Sets the maximum number of outgoing TCPIP channels to 25.

---

min_space <number>

This setting specifies the minimum amount of disk space DSMTP must have to work. If the available disk space falls below **<number>** megabytes, it will send an Email to the system administrator alerting them to the problem, and then it will refuse all connections. As soon as more disk space is made available, DSMTP will accept connections. Because of this, it is important that the problem be addressed as soon as possible. The default is 10, i.e. 10 Megabytes.

Example:

min_space 5

---

msg_filter <filename>

This setting tells DSMTP which file to use for its message filters. The file is a simple text file, containing any number of the following types of entry, one per line:
<rule> <part> <string>
<rule> may be either 'reject' or 'accept'.
<part> may either be 'body' or a header entry such as 'Subject', 'Return-Path'.<
<string> can be any text string, may include wild cards.

For example, a text file filter.txt might look like:
reject body eschatology
accept subject yak
reject subject *cat*arm*

The first line will cause DSMTP to reject any message with eschatology anywhere in the body.
The second line will cause DSMTP to accept any message with yak as the subject line.

The third line will cause DSMTP to reject any message with 'cat farming','cat charming' etc. in the subject line.

Note: DSMTP does a different type of search depending on whether it is checking a message header or the message body. Please see the notes below for specific differences.

If a message is rejected then DSMTP will give a message rejection code at the end of the DATA stage of the SMTP protocol, i.e.,
570 Command DATA Message rejected due to content.

Notes:

- The last applicable rule has priority. For instance using the above example, a message with eschatology in the body and a subject line reading "yak" will be accepted.
- You **must not** use wild cards in a body rule. e.g. you can not have a rule,
  reject body *.exe (**bad - do not use!**)
  it should be,
  reject body .exe
  or to be less general,
  reject body .exe"
  .This is because the search type done on a message body is different to the wildcard match done on a message header, as the need for processing speed is much greater.
- In versions prior to 2.8h, you must restart DSMTP for a message filter file change to take effect. For versions after 2.8h a tellsmtp reload is sufficient.
- For message body searches the <string> has to be **at least 3 charachters in length**. There is no such limitation on message header searches.
- See also the following FAQ,Can I filter messages based on the attachment name?

Example:

msg_filter /etc/dsmtp_filter

---

no_autohost <boolean>

If set to true, deactivates the auto adding of hosts to DSMTP's internal host_domain list. Default is false.

DSMTP keeps an internal list of all host_domain entries that DSMTP has read from dmail.conf. If DSMTP does a DNS MX lookup on a domain and finds that the end result points to itself (either IP address or domain) then it adds the domain that it looked up to its internal list of host_domain settings.

NB: it only adds Canonical or Primary names to its internal list. If the lookup resulted in a secondary domain that pointed to itself, dsmtp would not add the domain to its host_domain list and instead fail delivery and retry delivery later (hopefully when the primary server has come back online). For more information on this setting and its implications, please see
Your DNS (MX) Entries

Example:

no_autohost true

---

no_dotforward <boolean> (UNIX based platforms only)

If set to true, deactivates DSMTP's default behaviour of looking for a .forward file in the user's home directory as specified in the /etc/passwd file. Instead DSMTP will look for a .fwd file in the same directory as the user's drop file (including any directory hashing). The default is FALSE.

DSMTP has a method of creating forwarding rules where you create a file called, drop_file_name.fwd in the same directory as a user's drop file. To make DSMTP compatible with Sendmail systems, DSMTP will normally check to see if a user has any forward rules set up in a file called .forward in the user's home directory. DSMTP finds the location of the user's home directory by using the system call to retrieve that information from the /etc/passwd file. If the administrator sets up a domain specific passwd file with the domain_passwd setting, then DSMTP will check that file for the user's home directory.

For more information on .forward and .fwd files see, Forward Files in the 'Forwarding and Aliasing' section.

Example:

no_dotforward true

---

hide_rcvd_ip <ipaddress>

The specified IP address will be hidden by DSMTP in any Received header lines that it creates.

This option has been added so that DSMTP hides internal (made up) IP addresses on a LAN when it is acting as the gateway, so that they are not visible to the outside world.

NOTE: The IP addresses will be hidden in ALL received header lines, i.e. those added to BOTH incoming and outgoing messages.

Example:

hide_rcvd_ip 999.999.999.1
will ensure that DSMTP does not put the IP address, 999.999.999.1 in the received line of any messages that it receives.

---

oneline_stats <true/false>

If set true, DSMTP will use a single line per message in the new format below for its delivery message log.

Detailed statistics can then be found using the tellpop statistics command.

The new format is also more convenient for creating your own script or program for parsing the dsmtp delivery log.

NB: this setting was only added (and the tellpop statistics command) in version 2.9a.

DSMTP creates delivery logs in the log_path directory, with the filename of the form,

dmDDMM.log
where DD is a 2 digit day and MM is a 2 digit month.

The new syntax of the deliver lines is,
date_stamp fail/delivered date <date> from <address> to <address> size <size> messageid <message-id>

NB: we may add extra information to the line at any stage, hence the 'name value' style syntax. We will try to only add to the end of the line.

Example:

oneline_stats true

orbs_action <ban | reject | vanish>

If set, any of the listed actions causes DSMTP to lookup the ip address of the incoming connection on the ORBS database. If the sender is found to be from a listed site the specified action below is taken with the message.

Actions:

ban: causes the incoming connection to be dropped. DSMTP writes a message to the channel informing the sender that they have 'no permission to talk'.

vanish: the incoming mail is accepted if otherwise ok but is then deleted and is not recoverable.

reject: DSMTP will give 500 errors to all recipient lines.

NB: you may only specify ONE action

We have now added support for the MAPS RBL database which our customers inform us is more reliable as it is a list of known spamming IPs rather than just open relay sites. The setting for MAPS RBL lookups is,
maps_action <ban | reject | vanish>

You can set DSMTP to lookup incoming IP addresses on both databases, however the action specified for the ORBS setting will occur if the IP address is on both lists.

For more information on the ORBS database see,
http://www.orbs.org
and for MAPS RBL see,
http://maps.vix.com/rbl/

Examples:

orbs_action ban

orbs_action reject

orbs_action vanish

maps_action ban

maps_action reject

maps_action vanish

---

quiet <boolean>

This setting tells DSMTP to send a minimal amount of output to stdout (such as errors, and brief info on what it's up to). The information being logged to the log file remains unchanged. The **<switch>** parameter must be set to "true" to activate this option. The default setting is true.

Example:

quiet true

---

ras_domain <domain> (WINDOWS PLATFORMS ONLY)

Specifies a string containing the domain on which authentication is to occur.

If left blank the domain in which the remote access server is a member.

An asterisk specifies the domain stored in the RAS phonebook for the entry.

It **must** be a full domain name. See the FAQ for details on running a dialup mail server (NT only).

Example:

ras_domain my.pet.domain

(it is recommended not to use this setting)

---

ras_entry <string> (WINDOWS PLATFORMS ONLY)

The **<string>** parameter contains the **name** of the dial-up entry in the RAS phonebook that DSMTP should use when running in RAS mode.

Example:

ras_entry My Connection

---

ras_number <number> (WINDOWS PLATFORMS ONLY)

The **<number>** tells DSMTP which phone number to use when dialling up in RAS mode. This should match the RAS phonebook entry.

Example:

ras_number 5551234

---

ras_password <string> (WINDOWS PLATFORMS ONLY)

If the RAS entry requires a password, it must be entered here. This should match the RAS phonebook entry.

Example:

ras_password mypassword

---

ras_smtpip <ip number> (WINDOWS PLATFORMS ONLY)

Once DSMTP has established an RAS connection, it must contact an SMTP server to retrieve messages that are waiting for it. The **<ip number>** parameter tells DSMTP where it is.

Example:

ras_smtpip 1.2.3.4

---

ras_timeout <number> (WINDOWS PLATFORMS ONLY)

After contacting the SMTP server given in ras_smtpip messages will start arriving.

This is because after opening up the connection to the remote SMTP server, DSMTP sends the Extended SMTP command, ETRN domain, for all of the domains specified in the host_domain and vdomain settings in dmail.conf. The ETRN command causes the remote SMTP server to send all mail waiting in its queue for the specified domain.

If DSMTP does not receive any messages for **<number>** minutes, it will hang up the RAS connection. The default is 2 minutes. the minimum is 1 minute.

Example:

ras_timeout 5

---

ras_timer <number>

If using an RAS connection to get its mail, DSMTP will reconnect every **<number>** minutes. The default is 30 minutes, the minimum is 5 minutes. It would be disingenious to set this number to less than **ras_timeout** as DSMTP would never hangup the connection. Which somewhat defeats the purpose.

Example:

ras_timer 60

---

ras_username <string>

When establishing an RAS connection, a username is needed. This command tells DSMTP what that is.

Example:

ras_username Simon Travaglia

---

reject_no_reverse <switch>

Normally, DSMTP doesn't bother doing a reverse lookup on incoming connections, as it can take time. However, it is sometimes desirable to reject incoming mail from machines that fail a reverse lookup. To do this, turn on reverse lookups and then set **<switch>** to true.

NB: This setting will be ignored unless you have the lookup_names setting set to true.
DPOP also ignores this setting.

Example:

reject_no_reverse true

---

relay_to <domain>

The relay_to setting permits unconditional relaying to a particular domain.

Example:

relay_to somewhere.com

---

remind_timeout <number>

This setting specifies how long DSMTP should wait between events before alerting the system administrator of a recurring problem (see sysadmin . DSMTP will not send an Email if the event occurs within **<number>** seconds of the last one. This means that a frequently occurring problem will only generate one error message. The default is 3600 (or one hour).

Example:

remind_timeout 7200

---

robot_try <time>

This setting tells DSMTP how long it ought to keep trying to send input to a robot if it cannot send the whole message immediately. It will continue to try to send the robot more data until **<time>** seconds have elapsed since the last successful write. DSMTP will then give up trying to send more data, but will wait until the time specified by robot_wait before killing it. If the **<time>** parameter is larger than that specified by robot_wait the robot will be killed regardless of whether or not DSMTP was able to send the whole message to it. The **<time>** parameter is in seconds, with the default being 120 (or 2 minutes).

Example:

robot_try 10

When the write to the robot fails you will see a log line like, '*** Warning *** robot 11631 choked on 77 bytes' in dsmtp.log. If the robot_try time is reached then you will see a log line like, 'gave up trying to feed robot'.

---

robot_wait <time>

This setting tells DSMTP how long to wait after sending the message data to a robot before killing it. If the robot is still active after this time, DSMTP assumes that it has either hung, or is stuck in a loop, and terminates it. It can be set to any integer (negative numbers will have the same effect as 0, i.e. the robot will be killed immediately). Because there is no standard for robot implementation, they cannot be reliably assumed to terminate under all circumstances, so there is no option to switch off robot killing. The **<time>** parameter is in seconds, with the default being 600 (or 10 minutes).

Example:

robot_wait 10

---

rotated_logs <number>

DSMTP rotates it's logfile out once it reaches 1 megabyte in size. dsmtp.log is rename dsmtp1.log, dsmtp1.log is renamed dsmtp2.log, and so on. The total number of logfiles kept can be set by using rotated_logs. The default value is 4.

Example:

rotated_logs 10

---

shell_prefix <string>

Normally, DSMTP will run robots/autoresponder by directly forking and exec'ing the process. Sometimes this doesn't work, or doesn't have the desired effect. It is possible to set shell_prefix to fix this. If **<string>** is set to, say, "shell -c", dsmtp will run all autoresponders by forking, then exec'ing

shell -c "/usr/bin/whatever" instead of

/usr/bin/whatever

Example:

shell_prefix shell -c

---

show_8bitmime <boolean>

If set to true, then DSMTP will advertise its 8 bit MIME compatibility as part of the Extended SMTP welcome banner.

The default is false, i.e. do not advertise 8 bit MIME support.

When an email client or SMTP server opens up a connection to DSMTP, it sends the SMTP command HELO. For Extended SMTP servers, like DSMTP, it can send EHLO. In response to EHLO DSMTP sends a number of lines to tell the client what its capabilities are, e.g. here are the opening lines of an SMTP connection from the clients end,
220 tosh DSMTP ESMTP Server v2.5c
EHLO domainx.com

250-server.com. Hello domainx.com (161.29.2.46)
250-ETRN
250-DSN
250 HELP

If you want DSMTP to advertise 8 bit MIME as one of its capabilities set show_8bitmime to true, then DSMTP will show its 8 bit MIME capability, e.g.
220 tosh DSMTP ESMTP Server v2.5c
EHLO domainx.com
250-server.com. Hello domainx.com (161.29.2.46)
250-ETRN
250-DSN
250-8BITMIME
250 HELP

In versions prior to 2.5c DSMTP always advertised its 8 bit MIME compatibility, but we think that 8 bit MIME is a strange thing :-) and so DSMTP no longer does by default.

Note: If DSMTP advertises as 8 bit MIME and accepts a message in that format, then it cannot pass the message on to a server or client that is not 8 bit MIME compatible (e.g. if that server only accepted 7 bit MIME messages), it will simply bounce the message back to the sender.

Example:

show_8bitmime true
will make DSMTP always advertise its 8 bit MIME compatibility in its welcome banner.

---

show_ehlo [8bit],[vrfy],[auth]

DSMTP supports the ESMTP (Extended SMTP protocol) and as part of this if a client connects with the EHLO command DSMTP must respond by advertising which of the SMTP Extensions it supports.

The problem with this is that if advertised most clients tend to try to use the latest and greatest things even if they are not sensible. So this setting is provided to allow you to decide what DSMTP advertises as its capabilities.

DSMTP will always respond to EHLO with, ETRN,DSN and HELP, e.g.
250-netwin.co.nz. Hello bob.com (1.2.3.4)
250-ETRN
250-DSN
250 HELP
this setting lets you set what other things it advertises.

This settting is a comma separated multi value setting which takes any of the key words in the following table:

Keyword	Makes DSMTP Advertise
auth	AUTH PLAIN LOGIN
8bit	8BITMIME

vrfy	VRFY

NB: this setting only affects what DSMTP advertises not whether it supports each of the commands, e.g. whether you make dsmtp advertise VRFY or not, what DSMTP does in response to the VRFY command is set with the setting, fake_verify.

NB: If you have set,auth_allow relay, then you should not need to set, show_ehlo auth, as DSMTP will automatically advertise support for AUTH PLAIN LOGIN.

See also show_8bitmime which this setting can be used to replace.

Example:

show_ehlo auth,vrfy,8bit
will make DSMTP advertise AUTH PLAIN,VRFY and 8BITMIME as well as the normal ETRN,DSN and HELP.

---

smtp_port <number>

If used, this setting sets the port number which DSMTP will listen on to port **<number>** Any SMTP clients wishing to talk to DSMTP (including Tellsmtp) must use this port. Tellsmtp will look in DSMTP's default config file unless told otherwise with a -i setting on the command line, see the tellsmtp commands section. The default value is 25, of course.

Example:

smtp_port 1025
smtp_welcome;

---

smtp_welcome <string[$HOST,$DATE,$QFILES]>

A template for the SMTP welcome line, given by DSMTP when it first opens a connection.

Accepts \\r\\n delimited lines, so that it can be a multiple line response although this is not particularly recommended.

The <string> can be a template that accepts the following macros:

$DATE = the day's datestamp.

$HOST = the hostname of the machine DSMTP is running on.

$QFILES = the number of files in the queue at this point in time (not normally used, added by request for a customer's system where the sending device would destribute load based on this information).

The default at time of writing this is, 220 $HOST DSMTP ESMTP Mail Server

NB: You should start your line with the response code '220 '

Example:

smtp_welcome 220 Welcome to SMTP server, $HOST .
which might result in the welcome line,

smtp_welcome 220 Welcome to SMTP server, smtp.mymachine.com .

---

spool_dir <path>

Activates a mail spooling directory for incoming mail. DSMTP will deliver any mail found in this directory which is in a file with the ending, .msg.

If a message is written to file in this directory with the a filename ending in .msg then DSMTP will open the file, parse it and try to deliver it to the destination given in the message headers, namely the To: header.

The format of the .msg file should be a valid email message including headers, a separator blank line and a message body, e.g.

From: <user@domain1.com>
To: <user@domain2.com>
Subject: hello

First line of message body
Second line of message body

NB: the address(es) MUST be enclosed in angle brackets.

Webservers with CDONTS:

This setting allows you to send messages with CDONTS from web pages. In version 2.9a we have added checking for the CDONTS default file ending of .eml as well as .msg files.

Time delayed delivery:

If a message file name starts with a digit then DSMTP will try to work out a time setting from the file name. If it can then it will wait until that time (to the nearest 5 minutes) and then deliver the message.

If DSMTP can not work out the time setting it will deliver it immediately.

The format is,
YYYYMMDDHH.msg
where YYYY is a 4 digit year,
MM is a 2 digit month
HH is a 2 digit hour (24 hour clock)

E.g. 2000310813.msg DSMTP will try to deliver this message within 10 minutes of 1 PM, on the 31 August in the year 2000.

Notes:

- DSMTP looks for the To: header and if not found looks for the X-Rcpt-To: header.

- Multiple destination addresses should be given within angle brackets, so in this format,
  To: blah<address1>blah<address2>
  So addresses can be comma separated or whatever, but must be in angle brackets.
- The resolution is 5 minutes, so a message can take up to 5 minutes to be found by DSMTP. (this is because the file processing needed can be intensive)(NB: in 2.8n the resolution was 1 hour, and was set down to 5 minutes in 2.9a)
- Don't start message filenames with a digit else DSMTP may try and apply a time delayed delivery (in most cases it will realise and deliver immediately, but we can't garuantee that).
- This setting requires at least version 2.8n of DSMTP
- In 2.9a a file locking mechanism was added. If you set,
  spool_readyfile true
  in dmail.conf, then dsmtp will not read the .msg files until you create a file of the same name with the extension, .rdy. This means that you can create a .msg file and then when you have finished make a .rdy file with the same base name, and only when the .rdy file appears will DSMTP read the .msg file.
- Checking for messages with the .eml extension was added in 2.9a. NB: .eml message files can not make use of the time delay feature or the file locking feature.

Example:

spool_dir c:\dmail\spool_dir

---

sysadmin <email address>

This setting tells DSMTP who to contact if something terrible happens. Currently, those things are:

When DSMTP detects a mail-bomb incident

When DSMTP runs out of disk space (as specified by the min_space setting)

Also DSMTP automatically creates an alias for postmaster to this address (the postmaster alias is case insensitive and currently covers all domains).

It is highly recommended that this setting be included. Especially in the case of a disk full error, when DSMTP will cease to accept any incoming connections, it is most important that the problem be attended to. There is no default value.

This setting covers all domains and is not by default alterable by domain administrators.

Example:

sysadmin littlegreenman@area51.mil

---

tarpit_start <number>

This setting specifies the number of RCPT TO: lines at which DSMTP starts responding ever slower to further RCPT TO: lines in the SMTP protocol.

So if it is set to 7 for example, then when an SMTP session commences, DSMTP counts the RCPT TO: lines that the sender (i.e. an email client or another SMTP server) sends. On the 7th Rcpt To: line DSMTP will do the lookup on the email address, but then delay its response to the sender (e.g. 250 Rcpt User OK) by 1 second. After 3 more Rcpt To: lines it will delay each response by 2 seconds. Every further 4 lines received it will increase the delay by another second up to a maximum of 15 seconds.

This setting is an anti spam measure that is independent to the max_rcpts setting, but is intended to be used in conjunction with it. The slowing of the response is likened to being stuck in a tar pit, hence 'tarpit'. This is designed to be annoying to spamming robots as it clogs them up. If you simply reject a connection then it does not bother a spamming robot, but if you can slow it down then that should make your server less desirable as a spamming target.

This setting is not included by default, i.e. there is no point at which DSMTP responds deliberately slowly.

The setting tarpit_except is provided so that you can specify servers that should be an exception to this setting.

Example:

tarpit_start 5
will result in anything that tries to give DSMTP a message with more than 5 Rcpt To: lines, gradually getting a slower and slower response, starting with a one second delay on to the 5th Rcpt To: line.

---

tarpit_except <IPAddress>

This setting specifies any IP addresses which should never receive a deliberately slow response from DSMTP, even if the number of recipients in the session exceeds the tarpit_start setting.

This setting can take a comma separated list of IP addresses, and wildcard entries.

Example:

tarpit_except 1.2.3.4, 161.29.2.*
will result in connections from the IP address 1.2.3.4 and any IP address starting 161.29.2. not ever getting a deliberately slow response, i.e. being restricted by the tarpit_start setting.

---

tcp_max <number>

Specifies the maximum number of TCPIP channels that DSMTP listens on. This is the equivalent of DPOP's max_sessions setting. The default is 200 channels, or 200 concurrent incoming connections.

---

tcp_timeout <number>

This setting specifies how long DSMTP should wait for a response on a TCP/IP port before giving up. This only applies when DSMTP is actually expecting data. Once **<number>** seconds have elapsed, DSMTP closes the channel and discards the half-complete message (if receiving), or queues it for another try (if sending). The default value is 300 (or five minutes).

NB: Setting **<number>** to something small could have unpredictable results. Be **sure** that you set this setting longer than your **authent_timeout** setting!

DPOP has the equivalent setting, pop_timeout.

Example:

tcp_timeout 120

---

timezone <string>

On NT, the _timezone variable gives unreliable results. If the **timezone** command is present, DSMTP will insert the **<string>** parameter where appropriate. i.e. In date fields:

Fri, 19 Jun 1998 02:03:30 **<string>**

Example:

timezone +1200 NZD

---

unix_case <switch>

This setting tells DSMTP if it should use strictly case sensitive user lookups for **unix_user** authentication. Normally, if DSMTP fails to find a user, it searches again for a case insensitive match. If only one is found, it uses that, otherwise it fails. Setting **<switch>** to 'true' will disable this feature.

Example:

unix_case true

---

user_quota <switch>

This setting activates mailbox quotas. For each user, there should exist a file called <username>_inf. In this file should be a line 'quota <number>', where number is the user's quota (in bytes). **It is up to the system administrator to maintain the quota line in these files.** DPOP will maintain the 'used' line, and DSMTP will not allow mail to be delivered to a dropfile whose 'used' bytes is greater than 'quota'. This setting should be used in conjunction with drop_max

Note: If there is no 'quota' setting in a user's _inf file then, DSMTP will not impose a quota restriction on that user.

The user quota setting has been extended in Versions 2.4i and above to take a numerical value instead of the 'true' setting. The value specified is a default quota (in **kbytes**) for users who have no 'quota' line in their username_inf file. E.g.
user_quota 40
would result in any user without a 'quota' line in their username_inf file having a disk quota restriction of 40 kbytes.

Examples:

user_quota true
would turn on disk quotas for those users with a quota setting in their username_inf file (manually entered by the system administrator).

user_quota false
would turn off all disk quotas, i.e. DSMTP would not make any checks on the user's disk useage, ignoring all quota lines in username_inf files.

user_quota 90 (ONLY valid in versions 2.4i and above)
would turn on disk quota's for all users. DSMTP would check the username_inf file for a 'quota' line, and use it if found and for all other users a default quota of 90 kbytes will be imposed.

---

virtual_user_post <filename>
virtual_user_pre <filename>

virtual_user_post: Adds a sendmail style virtual user table, actioned immediately before lookup.

virtual_user_pre: Adds a sendmail style virtual user table, actioned before any other local-user rules apply.

Both of these settings allow you to specify a file in which you provide aliases for user database names. In sendmail this is the common way to add virtual domains.

We have added support for such a file to help with conversion to DMail. However we strongly recommend that you move to our 'vdomain' style virtual domains so that you can take advantage of the features that they offer now, and will offer in the future. We of course always continue to support these settings so you do not have to change.

The difference between 'post' and 'pre' is when DSMTP looks up a user in these files. The 'pre' file is used as soon as DSMTP gets a RCPT TO: line, before any other checks for mail redirection, including the forward settings and alias files. Whereas the 'post' file is only checked just before the user database lookup, **after** all other mail redirection.

This means that if a message is addressed to bob and a lookup of the 'pre' file results in that being an alias for the user, 1234, then dsmtp will forget about 'bob' and only look for dmail.conf settings (including in alias files) for user '1234'. '1234' is also the username it will lookup in the authentication module. Whereas if that same alias was in the 'post' file you could add forward and alias settings for user 'bob' and still have his database 'username' of '1234'.

The pre and post files have the same syntax of one 'rule' per line of the form,

<source><tab><destination>

The following are example syntaxes, along with effective changes to recipient addresses for each one:

	Syntax	Comment	Original Rcpt(s)	'changes to'	Altered Rcpt(s)
1.	user1@domain1 user2@domain2		user1@domain1	-->	user2@domain2

2.	user1@domain1 user2	No destination domain given, source domain implied.	user1@domain1	-->	user2@domain2
3.	@domain1 user2@domain2	Implicit '*' wildcard for username.	user1@domain1 user2@domain1 user3@domain1 ...	-->	user2@domain2
4.	@domain1 %1@domain2	Replace %1 with the user from the matching line.	user1@domain1 user2@domain1 user3@domain1	-->	user1@domain2 user2@domain2 user3@domain3
5.	domain1 user@domain1	If no @ in source, it's a fallback address	user1@domain1	-->	IFF user database lookup fails, user@domain1

NB: DSMTP offers all of the functionality of the virtual_user file with its other settings, see,
forward
alias_file
fallback_address

NB: Syntax 4 is also possible with forward settings in versions 2.8m and above, e.g.,
forward *@domain1 %1@domain2

---

warn_user <number>

This setting tells DSMTP when to alert the sender of a message that it is having trouble delivering the message. The **<number>** parameter is in retries, which occur every 2 hours. If the **<number>** parameter is larger than max_retry, DSMTP will only notify the user if it completely failed to deliver the message. The default value is 4 (which means the user will be alerted after roughly 8 hours). Delay warnings must be explicitly requested by the user by using the NOTIFY=DELAY ESMTP extension.

Example:

warn_user 5

---

use_forward_files true/false

This setting allows all checking of forwarding files to be turned off, so that DSMTP will not check for .fwd or .forward files.

The default is true.

For more information see,
forward files

Example:

use_forward_files false
would stop dsmtp from looking for any forward files.

Setting not found?!

You probably got here by using a link on our Complete Settings List page. That page lists all settings at the time of the latest compile, so we probably have not documented the setting you are looking for yet.

Please contact DMail Support with a list of any settings you want described and we will add them to these pages.

Back to Top of List

Products Downloads Prices Support Company

The Configuration File - dmail.conf

Following installation, various options and settings can be adjusted to tailor DSMTP and DPOP to your specific requirements.

Most configuration options for the three servers that make up DMail are held in the DMail configuration file, which is dmail.conf by default.

DSMTP and DPOP have **all** of their settings in dmail.conf, DList has most settings there too but also has a lists.dat file for list specific settings.

On this page ...

Links to the reference section setting lists ...

Format of the dmail.conf file

The initial settings in the configuration file are normally placed there by the DMSetup wizard. You may modify them at any time to fine tune DMail or to allow for changes to your system. The configuration file generated by the DMSetup wizard contains extensive comments and can be edited using a standard text editor.

Most settings in dmail.conf are specifed one per line and have the format,
name value
where 'name' is the setting name and 'value' is the setting value, commonly a single word or number. Some settings take multiple word values.

Nothing else should appear on the same line either before or after the setting. So DO NOT add comments onto the end of setting lines :-)

When appropriate settings can be entered multiple times on multiple lines, e.g.
setting1 value1
setting1 value2
...
Occasionally a setting can be specifed with a list of values separated by commas.

The reference section lists all available settings with their individual syntaxes.

See also, Notes on the dmail.conf file.

Where is the dmail.conf file located?

The location for this file is system dependent but typical values are:

- for most Unix variations

  `/etc/dmail.conf`
- for Windows 95 and NT

  `C: \winnt\system32\dmail.conf`

Notes on the dmail.conf file

- Multiple value settings should be entered as multiple lines in dmail.conf. A comma separated list of setting values may only be used when specifically stated in the reference section
- # and #include
- As some of the information stored in the configuration file may be sensitive it should normally not have world read access.
- Once you have made changes to a config file by editing it you must make the DPOP and DSMTP servers reload the config file for the changes to take affect. You can do this either by restarting them or by using the tellpop and Tellsmtp ' reload ' commands.
- Many settings can take the wildcard character, *. This will be specified in the reference section for any settings where it is allowed.

  Note that this only serves to make the entered value a simple string search. E.g.
  forward_from_ip 1.2.3.*
  specifies a value starting with '1.2.3.' and ending in anything.

  Our simple string search does provide support for negative entries (these will only make sense to use on some settings). E.g.,
  forward_from_ip !1.2.3.4
  tells dsmtp to allow forwarding from any ip address that is **not** 1.2.3.4

  DMail's wildcards are not a 'grep' syntax, e.g. this is **not** valid,
  forward_from_ip 1.2.3.[234]*

- In 2.8 versions and above, we have tried to make all ip address settings take a value range. An example of the syntax for this is,
  forward_from_ip 1.2.3.2-4
  which equates to 1.2.3.2,1.2.3.3,1.2.3.4

  You should not assume that a setting takes this syntax until you have tested it. Please contact, DMail Support if you would like confirmation on a setting.

NOTE : If you have modified a dmail.conf setting which is relevant to both DPOP and DSMTP, it is necessary to reload **both** servers individually with their reload configuration file commands, e.g. tellsmtp reload **and** tellpop reload. DMAdmin will do this automatically.

You can also use the DMAdmin GUI program to change configuration settings. It will make each of the servers reload the config file automatically after any changes are made.

Specifying a different configuration file

The DMail configuration file is dmail.conf by default, and is found in your system directory, e.g.
/etc/dmail.conf (UNIX based platforms)
\winnt\system32\dmail.conf (Windows NT)
\windows\system32\dmail.conf (Windows 95/98)

To specify a different configuration file for DPOP on startup use:

dpop -i another/path/fred.conf

or to specify a different configuration file for DSMTP on startup use:

dsmtp -i another/path/fred.conf

---

and #include

Within dmail.conf you can put a hash symbol, '#', at the start of any line that you want the DMail servers to disregard. I.e. the # symbol can be used for comments.

Note: Settings should be on a line by themselves. So you should NOT put comments on the end of lines that are settings! e.g. This is NOT allowed,
dsmtp_path /usr/local/dmail #path for DSMTP
but the following two lines are,
#path for DSMTP
dsmtp_path /usr/local/dmail
We realise that this can be annoying, and in general you will get away with it, but unfortunately there

are a few settings where it matters.

You can also use the hash symbol to include a file into the dmail.conf file, E.g.
#include /etc/domain2_forward_rules
will make DSMTP and DPOP open up the file, /etc/domain2_forward_rules, and read the lines from within it as if they were in dmail.conf itself.

NB: In versions prior to 2.7 versions you cannot put settings in a #include file that DPOP needs to see, as older versions of DPOP ignore such #include lines.

NB: Imapd needs to read your config file "dmail.conf" for every new connection. By default it ignores #include lines as it could take too long to read them all. You can force imapd to follow these using the setting "imapd_include_level". This defaults to 0, and specifies how many #include levels to follow. For example you may add

```
imapd_include_level 1
```

to the start of dmail.conf to tell imapd to follow #include's, specified in dmail.conf, but ignore #include's within the included files. Following #include's is only supported in imapd 4.3.3q or later. For further information see imapd.htm.

Configuration Settings Specific to DList:

This page details only those dmail.conf settings which apply specifically to DList.
For more settings see: Settings Common to all servers , Settings specific to DSMTP and Setting specific to DPOP

DList does not have a reload command, it will automatically notice any changes you make to dlist settings in dmail.conf.

NB: All DList settings in dmail.conf begin, dlist_

For more information on the configuration file see,
The Configuration File, dmail.conf

Setting	Example	Description
dlist_path	(none)	The directory for lists.dat and all DList work files, log file, and list archiving.
dlist_domain	host_domain	Sets the default @xxx address for messages from DList. Overriden by individual lists 'domain' setting (default is first host_domain settings).
dlist_loglvl	info	Logging level (debug,info,warn,error)
dlist_rotate	50000	Size of log file in bytes before it is rotated (default and minimum value is 60 kbytes)
dlist_smtp_host <domain[:port]>	127.0.0.1:3025	In version 2.5g and above, this setting lets you set the domain or IP Address of the SMTP server it should send mail to. Optionally you can add a colon followed by a port number that DList should connect to other than the default SMTP port 25. (On older versions, DList will ALWAYS send out messages to the ip address,127.0.0.1, but you can use this setting to set the port that it talks to.)

dlist_accept_from_request <true/false>	true	Added for Backwards compatibility. If true then accept messages from usernames containing '-request' (default is false).

Dmail.conf settings used by DList:

- host_domain
- dmail_path - no longer used, use work_path
- log_path
- dwatch_path
- drop_path
- work_path
- hash_spool

DList settings used by DSMTP or DPOP:

- dlist_path

Setting not found?!

You probably got here by using a link on our Complete Settings List page. That page lists all settings at the time of the latest compile, so we probably have not documented the setting you are looking for yet.

Please contact DMail Support with a list of any settings you want described and we will add them to these pages.

The DMail General Configuration Settings:

This page details only those settings in dmail.conf which are recognized by both DSMTP and DPOP, some marked thus (DL) concern DList as well. The other settings in dmail.conf are specific to a particular server. See: Settings Specific to DSMTP, Settings Specific to DPOP and Settings Specific to DList

NOTE : If you have modified a dmail.conf setting which is relevant to both DPOP and DSMTP, it is necessary to reload **both** servers individually with their reload configuration file commands, e.g. tellsmtp reload **and** tellpop reload. DMAdmin will do this automatically.

Compulsory Settings:

(If omitted, DMail may function unpredictably, or not at all.)

authent_method	sets the method used for user/password lookups
drop_path	specifies the directory to use for email drop files
host_domain	adds a domain name to the list of domains to be recognized as being local
work_path (DL)	specifies the directory for work files (also default for log and statistics files)

Optional Settings:

(These settings may be omitted, and all have reasonable defaults.)

authent_cache	Number of authentication requests to cache, External authentication only.
authent_domain	true for 'user@domain' instead of 'user' to be passed to the authentication process (external authentication only)
authent_timeout	Timeout (in seconds) for external authentication requests. (external authentication only)
authent_number	number of concurrent authentication processes to run (external authentication only)
authent_process	Specifies the executable to use for external authentication process(when authent_method external)
dwatch_path	specifies path for .pid and .wat files and other DWatch information
drop_prefix	If true then the virtual domain prefix is used as part of path for drop files.
hash_spool	Sets hashing method (0,1 or 2). Hashing is where drop_files are distributed across multiple directories.
log_level (NOT DL)	Specifies how much information to output to the logfile (one of error, info, debug).
log_path (DL)	Specifies where the server log files are to go.

lookup_names	Sets DSMTP and DPOP to do reverse DNS lookups
lowercase_username	Sets DSMTP and DPOP to be case insensitive for usernames and hence 'dropfile' names.
vdomain	Sets up a virtual domain for use with/by DPOP and DSMTP.
vdomain_passwd	Tells DSMTP and DPOP to use a separate passwd file for a vdomain (Unix)
vdomain_separator	Specifies the separator character to use with Virtual Domain prefixes.

Detailed Descriptions of DMail/DPOP configuration settings:

authent_cache <integer>

Sets the number of authentication requests to cache, for both DSMTP and DPOP. The default setting for DPOP is the minimum of the setting max_users and 1000. The default setting for DSMTP is 1000

Note the DPOP process size will grow with cache usage at about 100 bytes per cache entry so a 10,000 user cache will add about 1 megabyte to the process size. See the Authentication section for an overview of this topic.

Examples:

authent_cache 0	(no caching)
authent_cache 5000	(cache the last 5000 lookups)

DSMTP has a command,
tellsmtp clear_cache_all
for clearing its cache when you change the fwd="" field returned by an external authentication lookup - see Ext Auth FWD Field.

---

authent_domain <switch>

<**switch**> must either be true or false.

If true, this setting tells DSMTP and DPOP to include the domain name when sending a username to the external authentication process, i.e. they will send lookup user@domain. This is to permit the same username to exist on more than one domain. The default is false, i.e.only lookup user is passed. See the Authentication section for an overview of this topic.

This setting applies to all domains on both the DPOP and DSMTP servers. It cannot be changed by domain administrators.

Example:

```
authent_domain true
```

---

authent_method <method>

This setting specifies which type of user authentication DSMTP and DPOP are to use. **<method>** must be one of the following strings:

```
unix_user    Servers use the standard Unix user lookup(username/password)
             with or without shadow password file.

nt_user      Servers use the standard Windows NT user
             lookup(username/password).

external     Servers use an external process for user lookups.
```

The authentication process is used either for verification that a user exists (e.g. this is what DSMTP uses it for) or it is for a full authentication of a username and password pair (this is what DPOP uses it for).

In addition the 'external' authentication module can return more information on the user, e.g. their drop file name (including path), mail re-direction settings etc.

Note on the drop path:
the methods nt_user and unix_user mean that the servers will work out the drop path based on the settings, drop_path, hash_spool (directory hashing) and vdomain. The external authentication process can also request that the servers work out the drop path by returning the key word, "config" as the user's dropfile path. If the external authentication process returns a drop file name (including path) then it must do any directory hashing required.

If the "External" method is used, the authent_process setting must also be used, the authent_number setting may be used as well. See the
External Authentication of users
section for more details on how the external authentication process must behave.

The default is unix_user on Unix and nt_user on NT so that after installation all system users automatically have email accounts on DMail.

This setting cannot be changed by domain administrators, covers all domains and requires a RESTART of both DPOP and DSMTP if it is changed.

Example:

authent_method unix_user

or
authent_method external
authent_process process_file

---

authent_number <number>

This setting tells DSMTP and DPOP how many external authentication processes to run upon startup. It has no effect unless authent_method is "external". If the process being used for authentication has long delays then running several copies will allow authentication of a number of users to be overlapped. However, the caching of external authentication requests normally provides sufficient gains to avoid the need for multiple processes. So the default of 2 for both DSMTP and DPOP would normally be left unaltered.

If you wish to enter a setting to set the authent number to a different value (both servers will use the same setting) then base your setting on how fast the external program operates, and how many clients are expected to be connected at any one time.

The default setting is 2 for both DSMTP and DPOP. See the Authentication section for an overview of this topic.

We do not recommend a setting higher than 10, and suggest a value of 5 if you find that you do need it higher than the default of 2.

NOTE: changing this setting requires a RE-START of both DSMTP and DPOP. It applies to all domains on both the DPOP and DSMTP servers.

NB: increasing this setting on some platforms results in less file handles being available for TCPIP connections. See max_sessions.

```
Example:

authent_number 12
```

---

authent_timeout <seconds>

Timeout for external authentication requests. If an external authentication routine is being used for checking username/password pairs you may want it to timeout if no response is given for some time. This setting sets the length of time before DPOP and DSMTP time out on the external authentication routine.

NB: Before version 2.5d this setting only applied to DPOP.

You may need to set this longer if for example your user database is down for a short period each day or is regularly congested.

If an authentication module takes longer than this period to respond DSMTP will treat it as if it respond with the 'come back later' response, -DEAD.

If your authentication module does respond after the timeout period, DSMTP may think that it has given the response in answer to the next lookup. This is when it reports an 'out of sync' message.

In versions above 2.7q DSMTP has code to 'get back in sync' when this situation occurs.

We recommend setting this to 30 seconds if you are unsure of a value to use.

The default timeout is after 10 seconds.

Example:

authent_timeout 10

---

authent_process <executable>

This setting is used in conjunction with the authent_method setting. When the "external" method is used, the **<executable>** parameter is used to determine the name and location of the program to be used for external user authentication. It must include the full pathname and filename of the program. If authent_method is set to external then you MUST also set this setting (DSMTP will stop if you don't). See the External Authentication section for details of how an external_authentication process should work. An example nwauth.c program is supplied with the DMail distribution set.

Changing this setting requires a RE-START of both the DSMTP and DPOP servers. It applies to all domains and cannot be changed by domain administrators.

Examples:

If you wish to use NWAuth then assuming the default installation directories set this setting to,
authent_process \dmail\nwauth.exe
on Windows platforms and on UNIX based platforms,
authent_process /usr/local/dmail/nwauth

```
authent_process c:\mail\authent.exe
authent_process /usr/local/dmail/authent
authent_process /mypath/myauthent
```

---

drop_path <pathname>

There is **no default** for the drop_path so a setting for it **must** be included in the dmail.conf file.

Drop_path specifies the default directory for the mail drop files. These are the files that contain newly arrived mail messages for each user. Normally the drop file for a user Fred will be drop_path/Fred. This file will be appended to each time a new message arrives. The only time this directory will not be used for mail deliveries is when an external user authentication program (other than the one specified by the Netwin setting) is used. Even then, if the program returns "config" as its drop directory, **<pathname>** will be used.

Note these added features . . .
When **<pathname>** is used, DSMTP and DPOP will perform any directory hashing specified by the hash_spool setting before writing to the dropfile. On UNIX based systems, a leading ~ can be used to signify the home directory. If used, it will automatically be replaced by the user's home directory, so ~/inmail might become /usr/local/ralph/inmail.

The DMSetup installation prompts you for the drop_path setting with, for example, c:\dmail for NT and /var/spool/mail for Linux.

Examples:

```
drop_path ~/Mailbox           # might use /home/fred/Mailbox as fred's drop
                              file
drop_path /var/spool/netmail  # will use /var/spool/netmail/fred as fred's
                              drop file
drop_path c:\pop3\mail        # will use c:\pop3\mail\fred as fred's drop
                              file
```

---

drop_prefix <switch>

This setting determines how the drop file name is constructed when virtual domains are being used. Drop_prefix can be set to true or false. The default is true. When this setting is true the drop file name will include the virtual domain prefix.

NB: This setting and the resulting drop file name is **not** affected by, whether you are using IP based virtual domains or suffix based virtual domains or by the authent_domain setting.

For example consider the case when the following vdomain line is being used and a user fred is connecting to abc.com which maps to ip number 1.2.3.4

- vdomain abc 1.2.3.4 abc.com /mail/abc_com

This second item in this line sets the drop_prefix to 'abc'.

If drop_prefix is true the drop file used will be /mail/abc_com/abc_fred
(the underscore '_' is the default vdomain_separator setting)

While if drop_prefix is false the drop file used will be /mail/abc_com/fred

NB: the example above assumes that directory hashing is turned off (hash_spool 0). The hash_spool setting controls directory hashing which affects the path.

This setting applies to all domains and by default cannot be altered by domain administrators.

```
example:

drop_prefix false
```

For more information see, virtual domains and User Administration

---

dwatch_path <pathname>

This setting gives the directory for the .pid and .wat files. The .pid files are written by each of the servers, DSMTP, DPOP and DList on startup and removed on normal shutdown. They are used by DWatch to monitor their status. DWatch can restart each or all of them if necessary. The files with a .wat extension will also be stored in the same directory for each of the servers, e.g. dsmtp.wat, which contains information for DWatch on how to restart that server and how often before giving up etc.

When DWatch restarts a server it also renames the current log file to a 'ded' file and places it in the dwatch_path directory.

A useful trick to know is that you can stop most of the servers by putting a file in this directory named, server.exit, e.g. if dlist sees a file called dlist.exit in the dwatch directory, then it will shut itself down (and remove the exit file when it does so).

The default dwatch_path is dpop_path, however on Windows NT the DMSetup installation wizard creates a specific 'dwatch' directory and points the dwatch_path setting at it. See the DWatch section for more details.

This setting applies to all domains and by default cannot be altered by domain administrators.

```
Example:

dwatch_path /usr/local/dwatch
```

---

hash_spool <number>

Specifies how the mail drop file names are 'hashed' (distributed) into drop directories.

Hashing is used to make up for Unix slow linear directory search. It also avoids overly large directories on large email systems.

NOTE: This setting must be the same for both the POP and SMTP servers of your mail system. In the case of DSMTP and DPOP they will both use this same setting and therefore match.

The **<number>** parameter identifies which hashing method to use. There are currently three options:

```
0:   No hashing

1:   Add one extra directory level. The directories in the new level are
     named a-z, but note that the following is done to decide which drop
     files go in which of the 26 directories; sum the first four characters
     of the drop file name,modulo 26, i.e. any username beginning 'fred' will
     always end up in the same directory but it isn't named f for fred rather
     in this case it will be named b.

     E.g. fred's drop file might go in, \dmail\in\b\fred where the hashing
     has added the \b directory.

2:   Add two extra directory levels. A simple naming method is used; the
     first two characters are used to name the two intermediate directories,
     e.g. /f/r/fred
```

The directories generated by the hashing method are appended to the pathname specified by the drop_path setting. If the directories do not exist, then DSMTP or DPOP will create them. The default is "0", which implies no hashing.

This setting is global to all domains and by default cannot be altered by domain administrators. This setting applies to DSMTP, DPOP and DList.

Note: if this setting is altered you MUST restart both DSMTP and DPOP.

```
Example:

hash_spool 1
```

---

host_domain <domain>

This setting adds **<domain>** to a list of aliases for the machine running DSMTP. Any checks on a message's destination for (amongst other things) local delivery will also compare any names in this alias list with the destination.

NOTE: The first host_domain entry in the config file must be the host machine's **actual** name, i.e. the actual name that the machine has on the internet (or intranet) . Any messages sent from, or generated by the host machine will use this entry in the 'MAIL FROM' line to identify itself. So for a non-intranet server the first host_domain entry must be a domain name that can be resolved to an IP address by any machines wishing to talk to DSMTP.

Except for the first entry, the **<domain>** parameter may contain the wildcard character '*'. Note that *.domain.com does not include domain.com itself, but *domain.com does!

Any number of these settings may be used. There is no default.

Example:

host_domain my.resolvable.domain.com
host_domain *mydomain.com

---

log_level <level>

This setting determines what types of event information should be written to the logfile. There are three levels: error, info, debug. Error is the default setting, and the usual setting for operating in.

```
error:   the only information written will be errors, warnings, socket read
         and write information and minimal progress information.

info:    as well as the error information, this setting gives much more
         progress information, as well as file open and close calls.

debug:   as well as the info information, this setting gives a whole lot of
         internal status information, function calls...all sorts of stuff.
         However, it may slow down operation slightly and can produce large
         log files, so it is not recommended for normal use.
```

If DSMTP is crashing or doing very peculiar things, you should immediately ensure that it is running with the debug option.
So edit the log_level setting in dmail.conf to read,
log_level debug
then reload both dsmtp and dpop.

The most useful things to Netwin support are the config file and the log file. The log file is far more

useful if the log_level was set to debug.

If setting log_level to debug because of DSMTP problems you should also consider setting, log_data true. The log_data setting only applies to DSMTP and makes it log all of its TCPIP transactions (DPOP does this on debug log level anyway).

You should also consider setting, pop_event_log to make DPOP log POP3 events.

NB: This setting does not apply to DList, see the setting dlist_loglvl for setting DList's logging level. It does apply to both DSMTP and DPOP. The log file writing is cached so it should not detriment the servers' performance. (If you want dsmtp to not cache log entries then set, log_flush true.)

This setting is global (it does not apply to individual domains).

NOTE: Don't get scared by log lines that you see on info or debug level. They are generally of a technical nature, which can make them worrying :-) See, Deciphering Log Files for more help.

For more information see, Log Files.

See also,
Common Settings:
log_path
DSMTP settings:
max_loglen, rotated_logs
DPOP settings:
max_log_size, log_status

```
Example:

log_level debug
```

---

log_path <pathname>

This setting specifies the path for log files. The three servers record error and other informational messages into a file called servername.log, e.g. dpop.log for DPOP. Old versions of this file are called, in the case of DPOP, dpop1.log, dpop2.log etc. These files are stored in the work_path by default, but sometimes the system administrator may wish to keep all log files in one place. This can be absolutely anywhere, so long as it is a full pathname. By default, the logfile path is the same as the work path. Note that on Windows NT the DMSetup program creates a \dmail\log directory and sets the log_path setting to point at it.

In addition to dsmtp.log and dpop.log, both DSMTP and DPOP can be made to create statistics files, e.g. DSMTP creates a daily summary log file stored at <pathname>, with a name of dmddmm.log, e.g. dm2704.log. See Statistics Files in the Disk Use And Files section.

DWatch also stores a temporary log file, dwatch.log, at <pathname>.

This setting applies to all domains and by default cannot be altered by domain administrators. This setting applies to DPOP, DSMTP and DList, as well as DWatch (some DMail utilities do not make

use of it).

See also,
Common Settings:
log_level
DSMTP settings:
max_loglen, rotated_logs
DPOP settings:
max_log_size, log_status

```
Examples:
log_path      work_path
log_path      mypath/for/log/files/
log_path      C:\dmail\log
```

---

lookup_names <switch>

Set true if DPOP and DSMTP should lookup domain names for checking valid connections - (CAN BE SLOW). If this setting is true then when DPOP gets a connection from say 161.39.2.44 it will do a reverse DNS lookup to turn this into whatsit.company.co.nz. This is only useful for the following reasons:

1. if you want to have incoming connections referred to by their domains, i.e. logged with their domain name as well as their IP address.
2. if you don't want DSMTP to accept connections where reverse DNS lookups fail. To do this you should set
   reject_no_reverse true
   in dmail.conf as well, so that when a connection opens, if the reverse DNS lookup fails then the connection will be rejected. It is generally faster to restrict access some other way as reverse name lookups can be quite slow.

The setting for lookup_names is disabled unless **<switch>** is "true", the default is false.

This setting applies to both DSMTP and DPOP, it is applied to all domains and by default cannot be altered by domain administrators.

Example: To turn reverse DNS lookups on, set

lookup_names true

---

lowercase_username <switch>

If set true then DPOP will only work with lower_case usernames. So user BOB logging in will be assumed to be user bob, and his password will therefore be checked against user bob's password.

This means that drop files will also be forced to lowercase. This is how this setting effects DSMTP as well, as when true DSMTP will also only create lowercase drop files.

The default for this setting is false - i.e. usernames and hence dropfiles are case sensitive.

This setting affects both DSMTP and DPOP and is global across all domains. By default domain administrators cannot alter it.

Notes:

1. Previously this setting only applied to DPOP, i.e. versions before 2.4e
2. If your authentication module is not case sensitive (e.g. NWAuth, Windows NT, or your external auth module), then using this setting stops multiple case license user entries, e.g. you won't have three users bob, BOB and Bob.
3. DSMTP does not use this setting for deciding if incoming mail is for a valid local user. The lookup that it does on incoming mail for Bob@domain, is case sensitive, so it looks up Bob. If your password file only has a user bob, on NT it will succeed, on Unix it will fail.

   For this reason, we have the unix_case setting for Unix systems where you want such a test to succeed.

See Case Sensitivity for more details.

Example:
lowercase_username true

`vdomain`

Virtual Domain support. A number of vdomain lines may be used to specify domain prefixes, ip numbers, virtual domains and drop paths. This allows different drop paths and username prefix strings to be specified for users connecting to different virtual domains based on IP number or username suffixes. See the virtual domains section of the manual for details of setting up virtual domains for use with DSMTP and DPOP.

The default is no virtual domains.

Multiple lines (one for each virtual domain) are required.

These settings cannot be altered by domain administrators by default and they apply to both DSMTP and DPOP.

NOTE: after altering any vdomain setting you MUST restart both DSMTP and DPOP.

DPOP needs to be able to work out which virtual domain a user belongs to when they log in to check their mail. DMail provides two methods for doing this, virtual domains can be either **Suffix based** or **IP address based**.

For virtual domains based on IP address rather than username suffix use the following format:(where DPOP tells the difference between users by the IP address they log into to check their mail)

vdomain prefix IPaddress domain drop_path

where

prefix — This is the internal prefix that DSMTP and DPOP will attach to the start of the username, e.g. prefix_bob, for referring to the users of this domain. Externally this is only visible for setting up authentication, i.e. adding users, and naming drop files. See drop_prefix

IP address — This is the IP address that users from this domain MUST connect to in order to read their mail. This is only needed by DPOP, so DSMTP ignores it (DSMTP knows that incoming mail is for this domain because the mail is addressed to this domain). This is the KEY to how DPOP tells the difference between users on IP addressed based virtual domains, i.e. it detects the ip address they connect to and assigns them to a domain based on this information.

domain — This is the domain name for which these virtual domain settings apply

drop_path — This is the FULL path to the directory where users of this domain should have their mail stored (any hashing directories will be added onto this).

'~' can be used as the first character of the drop file path in the same way as it is used in a drop_path setting.

Examples: (assuming the default vdomain_separator of '_')

vdomain sal 1.2.3.4 sales.netg.com /var/spool/salesmail

will set up a virtual domain for the domain sales.netg.com. Users will connect to DPOP only on the IP address, 1.2.3.4 , with a normal username, e.g. bob's username is bob. The system administrator will have added a user bob to the authentication database with username, sal_bob and bob's mail will be found in the drop file, /var/spool/salesmail/sal_bob

vdomain dom1 161.33.2.44 ast.netg.com /var/mail/domain1

vdomain dom2 161.33.2.30 lnt.netg.co.nz /var/mail/domain2

For virtual domains based on username suffix rather than ip number use the following format: (where DPOP tells the difference between users by a suffix on the username that they use to login with)

vdomain prefix suffix domain drop_path

where

prefix — This is the internal prefix that DSMTP and DPOP will attach to the start of the username, e.g. prefix_bob, for referring to the users of this domain. Externally this is only visible for authentication, i.e. adding users, and naming drop files. See drop_prefix

suffix — This is the suffix that users will add onto the end of their usernames in their email client, e.g. bob will enter bobsuffix as his username in his email client, for the purposes of connecting to DPOP, to read mail. NB: this setting includes the separator character! This is the KEY to how DPOP tells the difference between users on suffix based virtual domains, i.e. users connecting (on any ip address) have their username checked for this suffix, and if found DPOP assigns them to this domain.

domain	This is the domain name for which these virtual domain settings apply
drop_path	This is the FULL path to the directory where users of this domain should have their mail stored (any hashing directories will be added onto this). '~' can be used as the first character of the drop file path in the same way as it is used in a drop_path setting.

Example: (assuming the default vdomain_separator of '_')
vdomain sal /sales sales.netg.com /var/spool/salesmail
will set up a virtual domain for the domain sales.netg.com. Users will connect to DPOP on any IP address with the username, user/sales, e.g. bob/sales . The sys admin will have added a user bob to the authentication database with username, sal_bob and bob's mail will be found in the drop file, /var/spool/salesmail/sal_bob

---

vdomain_passwd <domain> <path>

When using many vdomains with unix_user authentication, it is sometimes more convenient to maintain separate unix-style passwd files for each vdomain. This command permits that to be done.

The **<domain>** parameter must be a domain name, with no wildcards.

We STRONGLY RECOMMEND NOT using this setting on Windows based platforms - if you want a file type user database then use NWAuth.

Note: DSMTP and DPOP will create the drop files for all users in the domain which the passwd file is for with the same uid and gid as the passwd file itself. E.g. if you have a vdomain_passwd line for the domain domain1.com of /etc/passwd2 then all users in domain1.com will have drop files with the same uid and gid as the file, /etc/passwd2.

This setting has no default value and by default cannot be altered by domain administrators.

Example:

vdomain_passwd vdomains.r.us /usr/local/vdomsrus/etc/passwd

---

vdomain_separator

Virtual Domain separator. This setting specifies the type of separator to be used for separating the domain prefix string from the username or filename, when operating virtual domains. The default is '_'.
The domain prefix and separator only appear in the users drop file name (unless drop_prefix is false) and in the authentication database (to distinguish between the same username on two different domains).

NOTE: this setting together with the domain prefix set in the vdomain line DO NOT affect the username that the user logs in with (or their email address).

So if the separator was the default '_' and the vitrual domain prefix is 'dom1' for domain one, then

ralph from domain one becomes dom1_ralph. Similarly if the prefix for domain two is 'dom2' then user ralph from domain two becomes dom2_ralph in the password file. You can use this setting to change the separator to another character, e.g. '+', which would make the two examples just given become, dom1+ralph and dom2+ralph.

NB: This character is also used in naming the user's drop path, so think carefully of what character you use, for example * is not allowed as then you might get a drop file name of, /mail/dom1*amy, similarly '/' might cause ambiguity with a drop file name of /mail/dom1/amy.

Note also: you can NOT have any usernames with the vdomain separator character in them, in ANY of your domains if you are using virtual domains.

This setting is used by both DSMTP and DPOP and by default cannot be altered by domain administrators.

Examples:

```
vdomain_separator _         # e.g. dom1_ralph
vdomain_separator -         # e.g. dom1-ralph
```

---

work_path <pathname>

This setting specifies the directory for the working files, temporary files, data files etc. for all three servers, i.e. this includes DList. By default log and usage statistics files are also stored here. The default work_path for each server is the same as the server's work path setting, e.g. dpop_path for DPOP etc., however for DMail, a common DMail directory would normally be used for all three servers' work paths.

DMSetup suggests a default work_path of
work_path \dmail\work (Windows platforms)
work_path /usr/local/dmail/work (Unix based platforms)

Examples:

```
work_path drop_path                    # will set it to same as the
                                       drop_path setting
work_path /usr/local/dmail             # might be used on a Unix machine
work_path c:\mail\DMail\work           # might be used on an NT machine
```

Setting not found?!

You probably got here by using a link on our Complete Settings List page. That page lists all settings at the time of the latest compile, so we probably have not documented the setting you are looking for yet.

Please contact DMail Support with a list of any settings you want described and we will add them to

these pages.

Back to Top of List

Products Downloads Prices Support Company

Configuration Settings Specific to DPOP:

This page details only those dmail.conf settings which apply specifically to DPOP.
For more settings see: Settings Common to DPOP and DSMTP, Settings Specific to DSMTP and Settings Specific to DList

Note: After modifying a DPOP setting you should do a tellpop reload command or send a reload command to the DPOP server using DMAdmin.

Compulsory settings:

(If omitted, DPOP may function unpredictably, or not at all.)

NONE: specific to DPOP, but note there are some common compulsary settings

Common Optional settings:

(These settings may be omitted, and all have reasonable defaults. ! remember - you can get back to this list after using a hyperlink by using your browser's 'back' button)

bin_path	Path for user directories containing 'Bin' (i.e. DPOP's ordered drop file format) and msg index files.
bulletin_path	Directory to contain bulletin messages of form nnn.txt.
bulletin_from	Text to be sent as from line in all bulletins. Default is 'Email System Administrator'.
dpop_host	The TCP/IP address of the host DPOP is running on.
dpop_path	Installation directory for DPOP and manual pages etc.
manager_ip_address	IP addresses manager commands can come from (affects tellpop and DMAdmin etc.).
manager_ip_name	Domain names that manager commands can come from (affects tellpop and DMAdmin etc.).
manager_password	Password for valid manager commands (OBSOLETE after DPOP version 2.0 - no longer in dmail.conf).
pop_port	Allows the the port number pop3 clients will access to be set to a non-standard value. /td>
stats_path	Path for log files of per connection usage statistics, if set turns on statistics logging.
user_ip_address	IP addresses user connections can come from.
user_ip_name	Valid domain names user connections can come from.
valid_users	Valid usernames to get their mail here.

Obscure Optional settings:

(These settings may be omitted, and all have reasonable defaults.)

authent_cache	Now a common setting - Number of authentication requests to cache, External authentication only.
authent_timeout	Now a common setting - Timeout (in seconds) for external authentication requests.
bin_pfull	Criteria for compacting. Bins with less than bin_pfull get compacted.
check_gid	(UNIX only) Set if gid for drop files should be checked before DPOP accesses drop or bin files. Default is no check and set no group access.
check_owner_disable	(Unix only) Set if user id for drop files should not be checked. Default is to check.
crlf_stored	Is cr and lf stored at end of lines on this system.
drop_ext	Change-over compatibility; Drop file extension. Default is none.
drop_kill	Change-over compatibility; File related to drop file to remove after drop file has been burst.
drop_old	Change-over compatibility; Process temporary files left by previous popper.
drop_prefix	Common Setting!
drop_users	Specifies users who need to be able to read mail from another popper as well as DPOP (i.e. convert from bin file back to drop file).
log_status	Time (in mins) between logging DPOP status to log file.
lowercase_username	Now a common setting to both DSMTP and DPOP (2.4e and above).
lowercase_password	If true DPOP will always set passwords to lower case before using them for authentication (auth module must have lower case passwords or be case insensitive).
max_log_size	Max size, in bytes, for DPOP log file before renaming and starting new one.
max_sessions	Limit on number of concurrent sessions connected at one time, DPOP will try to make this many available.
msg_separator	Change-over compatibility; Message separator character if not defined then we don't use one.
pop_timeout	How long DPOP waits (in seconds) before assuming a connection has gone and close TCPIP channel.
pop_event_log	Allows you to specify which POP events get logged: last,uidl,burst,list,listn,stat,user,pass,quit,retr,dele,rset.
retr_chunk	Max bytes to send in one chunk for retrv command. Default is 10000.
slave_number	Sets the number of DSlave processes (sub processes of DPOP) for handling burst of large drop files (we recommend 4).

slave_burst_size	Burst drop files of this size (in bytes) or larger with a DSlave process.
slave_timeout	Timeout (in seconds) for commands (e.g. burst) given to DSlave processes. Default 100 seconds.

DPOP configuration settings

---

bin_path

This setting specifies the directory for user work files. That is the username.bin sub directories. These contain bin files for processed messages and message index files. The bin_path defaults to the same as the work_path. It is often set to be the same as the drop_path. In this case a user with a drop file /var/mail/fred would have a bin directory /var/mail/fred.bin

```
Examples:
bin_path        drop_path
bin_path        /usr/local/bins
```

---

bin_pfull

This setting controls the criteria for compacting message storage bin files. Files with less than bin_pfull bytes used / total size are compacted when a user connection is closed. The default setting is 0.5

```
Examples:
bin_pfull 0.9                 # This would make compacting
                              happen more often
bin_pfull 0.1                 # This would make compacting
                              happen less often
bin_pfull 0                   # This would stop bins ever being
                              compressed
```

---

bin_path

Path for user directories containing the bin and message index files. Each user's bins and msg index files are stored in a subdirectory /username.bin below the bin_path. The default is the same as the work_path. The usual choice is to either store the bin files in the same place as the drop files or all together in the DPOP work_path.

```
Examples:
bin_path work_path          # will set it to same as work_path
bin_path drop_path          # will set it to same as drop_path
bin_path path/for/binfiles  # will set it to path/for/binfiles
```

---

bulletin_from

In addition to mailing lists DMail provides a bulletin facility. This setting determines what text will be sent in the from field of bulletin messages. The default setting is Email Administrator.

- A bulletin directory is specified in the dmail.conf file with a setting: e.g. bulletin_path /var/mail/bulletins
- This directory contains files with names of the form nnn.txt, where nnn is the bulletin id number
- Any user connecting to DPOP will receive any of these files they have not already seen as an email message
- For each user DPOP stores the the id number of the last bulletin seen by this user.

Click here for more detail: Description of Bulletin Facility

```
Examples:
bulletin_from    Your lord and master
```

---

bulletin_path

In addition to mailing lists DMail provides a bulletin facility. This is useful when you need to send a email message to all users but without the overheads of duplicating and sending the same message to all users. The bulletins are organized as follows:

- A bulletin directory is specified in the dmail.conf file with a setting: e.g. bulletin_path /var/mail/bulletins
- This directory contains files with names of the form nnn.txt, where nnn is the bulletin id number
- Any user connecting to DPOP will receive any of these files they have not already seen as an email message
- For each user DPOP stores the the id number of the last bulletin seen by this user.

Click here for more detail: Description of Bulletin Facility

```
Examples:
bulletin_path    /var/mail/bulletins
```

---

check_gid Unix Platforms Only

Set if Unix group id for drop files should be checked by DPOP and set by DSMTP when writing to drop files.

The default is for no check to be done and the drop file's group ID to be set to the gid for the user (or the gid for the uid which is returned from the external authentication routine). Note: if check_gid is not specified then the file protection on the drop file is always set to

no group access (0600) by DPOP.

DPOP always sets the owner of the drop file to the user.

NB: If set, DSMTP sets the group ID for the drop file to the specified GID if it exists.

No check is ever made on Windows platforms.

NOTE: you must RE-START DPOP after changing this setting, a tellpop reload is not sufficient.

This setting may be required so that your SMTP server can access the drop files - this should not be necessary for DSMTP, but may be for non-Netwin SMTP servers.

```
Example:
check_gid        mail    (would check group was mail and set
                         group to mail)
check_gid        netmail (would check group was netmail and set
                         group to netmail)
no check_gid setting     (would set to no group access)
```

---

check_owner_disable Unix Platforms Only

DPOP normally checks that a user's drop file is owned by the user, if it is not it reports an error in bursting the drop file. If you do not wish DPOP to make this check, set check_owner_disable to true. Think carefully before setting this!

No check is ever made on Windows platforms.

Example:

check_owner_disable true

---

crlf_stored

This should be set true if cr and lf are normally stored at the end of each line in a text file. DPOP will normally choose the correct default for the system it is running on. Thus the default is true for Windows NT and false for Unix. In some rare cases it may be necessary to use this setting explicitly.

```
Examples:
crlf_stored    true
crlf_stored    false
```

---

dpop_host

The TCP/IP address of the machine DMail/DPOP is running on.

- Tellpop commands will try and connect to DPOP running on the machine specified

- There is no default for this setting so it must be specified in dmail.conf, in recent versions (2.5f and above) the default is the first host_domain setting, so you do not need to set this setting.

```
Examples:
dpop_host    fred.dr.ac.nz
dpop_host    161.39.22.44
```

---

dpop_path

Specifies the installation directory for DPOP. The manual and utility applications are stored here. The default when DPOP is running as part of DMail is a directory called dmail, e.g. for Unix the default is /usr/local/dmail and the default for NT is C:\dmail. The DMSetup installation wizard creates the dmail directory and points all of the server's home directories at it, i.e. dpop_path, dsmtp_path and dlist_path are all made to point to the directory called dmail by default.

```
Examples:
dpop_path /path/for/dpop
dpop_path d:\network_apps\dmail
```

---

drop_ext

If drop filenames have a standard extension then it can be specified with this setting. For example if all drop files had a .mail extension then you would use the setting drop_ext .mail

The default setting is no extension. If using DSMTP then drop_ext is not required.

```
Example:
drop_ext                                    .mbx
```

---

drop_kill

Specifies a file related to the drop file to delete after processing the drop file.
On some systems some form of index file has to be killed after a drop file has been processed. For example for user Fred there may be a Fred.idx which needs to be killed. The default setting is none.

```
Example:
drop_kill .idx (will delete Fred.idx after processing drop file
Fred)
```

---

drop_old

Process temporary files left by previous popper.This setting normally only needs to be set for a short time after changeover to DPOP from some other popper. It tells DPOP to look for and process files with a particular extension before processing the normal drop files. For example a previous POP server may have copied some messages into a temporary file called username.pop and these will need to be processed before the normal drop file is processed.
The default is none, i.e. don't process any old temporary files.

NB: In version 2.7m if the drop_old setting contains a seperator DPOP will use the given path rather than thinking it is a suffix and tacking it onto the drop file name. So DPOP will read drop files from another path given by this setting, as well as reading its ones in the hash directory. This lets you move to another hashing method. NB: this does not mean old bin files will be checked so you have to drop all users first, e.g. tellpop drop_all.

Examples:

```
drop_old .oldone (Before processing drop file for user fred process fred.oldone)
drop_old .~.pop  (Before processing drop file for user fred process file
                  .fred.pop)
```

---

drop_users

NB: Do NOT use this setting unless you understand it! :-)

A wild card list of users who access their mail from pop3 AND the Unix command line. For these users we have to convert any undeleted mail back to a drop file after each connection. This degrades pop3 performance for those users so use sparingly. Note: This is not needed if users ONLY read mail from a Unix command line client and never get their mail through DPOP or if they always collect their mail using DPOP.
The default is none.

Examples:

```
drop_users              unixman
drop_users              fred,john,bill
drop_users              uni*,JSmith
drop_users              billsmith
```

---

log_status

Sets the time between logging DPOP status to the log file in minutes.
The default is 10 hourly, i.e. 600

Example:

```
log_status 120      (for logging every 120 minutes)
```

---

lowercase_password

If set to true DPOP will always set passwords to lowercase so that case is essentially ignored. The default setting is false - i.e. passwords are case sensitive.

```
Example:
lowercase_password true
```

---

manager_ip_address

The setting specifies a wildcard list of IP addresses which manager commands can come from. Specified in Netwin wildcard format: (*-wild, !-NOT, separated by commas. NO SPACES ALLOWED). Often set to * as manager commands must also contain the manager password. When Tellpop or the DMAdmin package are used remotely a challenge-encrypted-password response sequence is used. It is recommended that you include the '127.0.0.1' address and your server's address in order for the remote manager tools to be allowed to connect.

Note: This setting will not allow access unless either of the user_ip_address or user_ip_name settings also allow access.

```
Examples:
manager_ip_address      *
manager_ip_address      127.0.0.1, your.local.machine.name
manager_ip_address   161.29.2.*
manager_ip_address   161.29.2.37
```

---

manager_ip_name

The setting specifies a wildcard list of domains which manager commands can come from. Specified in Netwin wildcard format: (*-wild, !-NOT, separated by commas. NO SPACES ALLOWED). Often set to * as manager commands must also contain the manager password. When Tellpop or the DMAdmin package are used remotely a challenge-encrypted-password response sequence is used. It is recommended that you include the 'localhost' name and your server's name in order for the remote manager tools to be allowed to connect.

Note:

1. This setting will not allow access unless either of the user_ip_address or user_ip_name settings also allow access.

2. This setting can only work if you have allowed reverse name lookups, i.e. lookup_names true.

```
Examples:
manager_ip_name      localhost, your_machine_name
```

```
manager_ip_name      *
manager_ip_name      *.fred
```

---

manager_password

Specifies the password to be used for manager commands. This setting is now obsolete (from DPOP 2.0) Use command Tellpop password xyz instead. The password is no longer sent as clear text and is stored encrypted in a separate file tellpass.dat

```
Example:
#manager_password xyz
```

---

max_sessions

This setting can be used to limit the number of concurrent POP users. The default setting is 200. The number of concurrent connections is sometimes limited by other factors such as the number of available file handles. DPOP needs several file_handles per connection and on some versions of Unix file_handles per process are limited to 256. On NT and other versions of Unix this limit may be many thousands.

NB: DSMTP has its own setting to limit TCPIP channels used, tcp_max.

NB: you must be careful when chaning these settings that your OS can support the number of file handles needed by them. You should figure on 2 file handles being needed per TCPIP connection in both DSMTP and DPOP and another 2 for every authent_process that they run. Also both processes need about 10 file handles for things like log files.

On UNIX platforms you usually set the number of file descriptors available with a setting like, ulimit -n x, so you may need to set this in the startup scripts, dpop_start.sh and dm_start.sh.

When DPOP starts up on log_level debug, it will log a series of messages where it calculates the number of file handles it thinks are available. It then will modify your settings if they are set to high for what it can get. However it cannot check for file handles being used by other processes at peak times so be careful when setting this setting to higher values - you should monitor the effects.

File handle problems are often evident by errors about 'SELECT CALL' failures in the dpop and dsmtp log files.

```
Examples:
max_sessions    99
```

---

max_log_size

This sets the maximum size, in bytes, for the log file before it is renamed and a new one

started.
The default is 3,000 Kbytes (3Mb)

```
Example:
max_log_size   3000000  (for a maximum log size of 3Mb)
```

---

msg_separator

Message separator character. This allows you to make DPOP read non 'sendmail' type drop files which use a separator character to separate messages within the drop file. If not defined then we don't use one and depend on a blank line followed by a valid 'From' line as a message separator (as per Sendmail drop file format).

Also, the syntax x\n for this setting can be used to imply that separator character stuffing should be carried out, i.e. email message lines in the drop file which start with the character x will be stored as xxtext and that a line which contains just the character x is used as a message separator. For example the setting msg_separator .\n would mean lines starting with a dot would be stored as ..line and a line containing just . would signify the end of a message.

Examples:
msg_separator .\n (separator is '.' and dot stuffing should be done)
msg_separator |

---

pop_event_log

This setting specifies the POP3 events that should be logged to dpop.log. The default is that no commands are logged. NB: dpop will not necessarily log exactly the command given, it will log the command name and any useful debugging information from the command.

The setting syntax is,
pop_event_log <command_name>,<command_name>,<command_name>...
where command_name can be any of,
last,uidl,burst,list,listn,stat,user,pass,quit,retr,dele,rset

It is intended that you will use this setting in addition to log_level debug.

This setting was added to dpop in DMail version 2.8b

Example:
pop_event_log user,pass,quit
might result in the following log lines,

29 09:55:59 [0] Debug: >Got {USER test0}
29 09:55:59 [0] Debug: >Got {PASS xxxxxx}
29 09:55:59 [0] Debug: >Got {QUIT }

where '[0]' indicates the TCPIP channel that the event occurred on.

pop_port

This sets the port number that DPOP will accept POP3 client connections on. The default is 110. This can be set to something other than 110 for testing DPOP while leaving another pop server running on the normal POP3 port.

```
Examples:
pop_port 1100
```

---

pop_timeout

How long we wait (in seconds) before assuming connection has gone. If this setting is too long then a session left running will prevent the user from connecting from elsewhere. If it is set too short then the connection could be terminated while the user is just thinking what to do next. The default setting is 600 seconds or 10 minutes NB: this is the minimum value required by the RFC1939, so you should not set it lower than this except for testing.

NB: Setting this setting to something small could have unpredictable results. Be **sure** that you set this setting longer than your **authent_timeout** setting!

DSMTP has the equivalent setting, tcp_timeout.

```
Example:
pop_timeout    1200
```

---

retr_chunk

Sets the maximum number of bytes to send in one chunk for the retrv command.
The default is 10000, i.e. 10kb.

```
Example:
retr_chunk 5000     (for a maximum chunk size of 5kb)
```

---

slave_number

Specifies the number of DSlave processes to use for handling bursts of large drop files. The default is 0, i.e. no slave processes will be used. If you have a large number of concurrent users and some with large drop files or a slow file system then it is generally worth setting up several slave processes.

In version 2.5g and above DMSetup will set this to 4 when installing DMail.

```
Examples:
slave_number 2
```

```
slave_number 10
```

---

slave_burst_size

This setting determines when DPOP will use a slave process for bursting mail drop files. DPOP will use a slave process for any drop files larger than the specified size in bytes. The default for this setting is 1000000, i.e. 1Mb. Note that DPOP will only use slave processes if the setting for slave_number is 1 or more.

In version 2.5g and above DMSetup will set this to 500000 (500 Kbytes) when installing DMail.

```
Examples:
slave_burst_size 1
slave_burst_size 10000000
```

---

slave_timeout

Sets the timeout value for slave commands such as burst drop file. The default is 100 seconds. If the timeout is exceeded then the slave process is killed and a new one started. The operation that timed out is aborted.

```
Example:
slave_timeout 123
```

---

stats_path

Specifies the path for files containing a log of per connection usage statistics. One file is produced each day and they can be analyzed and summarized with the Tellpop stats command. The default is to place these stats files in the dpop_path. If no stats records are required then a blank path can be specified.

```
Examples:
stats_path              # implies don't log usage statistics
stats_path c:\pop_stats  # place stats files in c:\pop_stats
```

---

user_ip_address

Specifies IP addresses that user connections can come from. Specified using Netwin wildcard format: (*-wild, !-NOT, separated by commas. NO SPACES ALLOWED). The default setting allows connection from any ip address. This is only one of several ways of limiting access, see Controlling Access

Note: This setting applies to any incoming TCPIP connections, even manager (DMAdmin or tellpop) commands, so you should include, 127.0.0.1

```
Examples:
user_ip_address 127.0.0.1
user_ip_address 130.123.*.*
user_ip_address 130.123.24.*,!130.123.24.25
user_ip_address 130.123.*.*
user_ip_address *
```

---

user_ip_name

Specifies valid ip names user that connections can come from. Specified in Netwin wildcard format: (*-wild, !-NOT, separated by commas. NO SPACES ALLOWED). The default setting allows connections from anyone. This is only one of several ways of limiting access, see Controlling Access

Note:

1. This setting applies to any incoming TCPIP connections, even manager (DMAdmin or tellpop) commands, so you should include, localhost, and probably your machine name, in order for DMAdmin to work.

2. This setting can only work if you have allowed reverse name lookups, i.e. lookup_names true.

```
Examples:
user_ip_name     localhost
user_ip_name     massey.ac.nz
user_ip_name     massey.ac.nz,otago.ac.nz,fred.john.bill
user_ip_name     *.ac.nz,bill.*.nz
user_ip_name     *
```

---

valid_users

Wildcard list of valid usernames. Only users whose usernames match will be able to connect to DPOP. The list is specified in Netwin wildcard format: (*-wild, !-NOT, separated by commas. NO SPACES ALLOWED)
The default allows any username to connect. Note this is only one of several ways of limiting access.

```
Examples
valid_users     *, !*smith*,fred,john,bill
valid_users     *
```

Setting not found?!

You probably got here by using a link on our Complete Settings List page. That page lists

all settings at the time of the latest compile, so we probably have not documented the setting you are looking for yet.

Please contact DMail Support with a list of any settings you want described and we will add them to these pages.

(Back to Top of List)

Search: Enter Word(s) to find

Submit

Free Trial Updates

License Prices

Products Downloads

Support Company

The Basics of DMail

- Email Servers

This manual describes DPOP a POP3 mail server, DSMTP an SMTP mail server, and DList an email list server. Together they provide a complete email server solution for small and large internet and intranet service providers.

DPOP is a highly efficient scaleable POP3 mail server which operates as a single process and can handle a large number of concurrent email client sessions. It is a drop-in replacement for popper and other POP3 servers. It provides an efficient and scaleable solution for email providers. It places no unnecessary limits on concurrent connections. Smaller systems may optionally limit concurrent connections to a few hundred. The total number of user or client email accounts may be limited by the type of license. (for example <5, <50, <500 or unlimited versions are available) Systems with 500,000 client accounts and many hundreds of concurrent connections are quite achievable even with relatively modest hardware.

DSMTP is an efficient and scaleable SMTP mail delivery server. It operates as a single process and can handle a large number of concurrent connections. It is a drop-in replacement for sendmail. It works with DPOP to provide a complete email solution. When a user sends email their email client package connects to an SMTP server (DSMTP). If the email is for a local user DSMTP writes directly to the dropfile. If the email is for a remote site then DPOP will connect and transfer the email as required.

DList is an easy-to-use and efficient email list server. It allows mailing lists to be easily set up and maintained. It can be entirely controlled via email. A Web based manager for the DList administrator and email list moderators will be available shortly.

Table of Contents:

Introduction to Internet Mail Servers

The provision of internet email depends on a number of software components working together. These days most email users access email by running an email client application such as Eudora or Pegasus mail on a personal computer. This client application talks to two server applications which are normally located on a host machine which is run by an internet service provider or local computer service group.

To provide internet email to computer users, three applications must work together; the client, the collection server and the delivery server. The collection server is what an email client connects to in order to collect or read the user's email. The delivery server is what the client connects to when it wants to send a message.

An email client sends outgoing mail to an SMTP server which transfers the mail to other SMTP

servers and eventually one of them stores it on the machine from which the client will read it. DSMTP is Netwin's SMTP server product. It would often replace applications such as sendmail.

The most common collection servers use the POP or POP3 protocol when interacting with the client. Such a server is referred to as a POP server. DPOP is Netwin's POP3 server product. It would often replace applications such as popper, or qpopper.

A fourth optional component of most email systems is an email list server. Email lists are lists of email users that you can send messages to as a group. For example you might have a group of users all part of the social club organizing committee. When discussing the organization of social events via email you need to be able to send messages to all of the committee members. This is sometimes handled by a simple list of email addresses stored on the user's PC within the email client software. This is very inefficient in that the same message is sent repeatedly form the client PC to the host server and then perhaps from there to various other sites. A more efficient method is to have a list server running alongside the mail server. It accepts messages for a pseudo user named social_cmty, for example, and passes them on to all users listed on its social committee list. This avoids the need to repeatedly send messages, and avoids the need for the users to all maintain a list of people on the committee. It can also provide other features such as an archive of emails to the list, a set of shared files which list members can download etc. DList is Netwin's email list server product. It would often replace such applications as majordomo.

Traditionally the various parts of the email server family, particularly on Unix platforms have not been designed with efficiency in mind, email was traditionally short test messages and the numbers of users small. Each connection was a sort of pseudo login starting a new process up and interacting with the client software. With much larger numbers of users and larger email messages which often contain file and image attachments more efficient systems are required. The management of the components of an email server system and the gathering of usage statistics was also problematical. The setup and maintenance of mailing lists was often sufficiently complex as to deter many small groups of users from using real mail lists. The Netwin suite of mail products; DSMTP, DPOP and DList were designed to overcome these limitations and to allow the hosting of complete mail systems on quite modest hardware. The ease of management of email servers was also paramount in the design.

DNS (MX) Entries

You add DNS entries that allow other servers on the internet to resolve your IP address from your domain name.

If you have just one domain then you will probably only add a simple DNS entry, called an A name. A names resolve a domain to an ip address, e.g. netwinsite.com to 207.230.97.10 . Try going to a command prompt and entering,
nslookup <cr>
netwinsite.com <cr>
exit <cr>
hopefully it will give you our ip address.

If you have multiple domains on your email server then you will probably add MX records for your

virtual domains. These return your main domain which in turn resolves to the IP address of your email server. So again using the nslookup program try,
nslookup <cr>
set type=MX <cr>
netwinsite.com <cr>
exit <cr>
hopefully it will give you the the domain that hosts our site in the United States.

Other things you might want to read up on are Cnames and rotating DNS entries.

Here is a good place to start, DNS Resources Directory.

Stopping, Starting and Re-Starting the DMail Servers

After changing a configuration setting in the configuration file, dmail.conf, the DMail Servers need to be notified.

Reload:

Most re-configuration changes, e.g. adding forward settings or aliases, to the DMail servers (dmail.conf changes) only require the servers to be instructed to reload the configuration file.

A reload instructs the server to re-read the configuration file without going offline. DSMTP and DPOP can be 'reloaded' separately and DList does not require reloads. To do this you can:

1. enter the reload command at a command prompt using the command line programs tellsmtp and tellpop:

tellsmtp reload
and
tellpop reload

2. Issue the command(s) locally or remotely with the DMAdmin Windows GUI administration tool, by selecting the 'Reload configuration files' command to be sent to each of DSMTP and DPOP. The remote administration using DMAdmin from a windows box is possible no matter whether the servers are running on Windows or a UNIX based platform.

Note: A reload can take up to about 30 seconds for a very large dmail.conf file and require substantial processor useage. So to avoid performance loss with very large dmail.conf files (e.g. 500+ domains) you should probably automate your reloads to a half hourly or hourly interval and notify domain administrators and customers of this buffering of changes.

Re-Start

Here are the suggested ways to restart a server for each platform. More details on starting and stopping the servers is provided in the sections below.

Windows Platforms:
a) CLick 'Stop All' button followed by the 'Start All' button in DMAdmin.
b) CLick on the individual server's 'Start server_name'/'Stop server_name' button on the appropriate server tab in DMAdmin.

UNIX based Platforms:
For DSMTP and DPOP enter,

tellsmtp shutdown
tellpop shutdown

followed by,

/usr/local/dmail/dm_start.sh
/usr/local/dmail/dpop_start.sh

You do not need to restart DList after changing dmail.conf or lists.dat settings.

Stopping a Server:

1. enter the shutdown command at a command prompt using the command line programs tellsmtp and tellpop:

   tellsmtp shutdown
   and
   tellpop shutdown
2. Issue the command(s)locally or remotely with the DMAdmin Windows GUI administration tool, by clicking the 'Stop server' button on the appropriate server tab.

To stop DList on UNIX based platforms simply **kill** the dlist process.

Problems Stopping a Server?

On Windows:
You will find that you cannot 'end' DMail server processes in the task manager because they have been started by the DWatch program, which is probably being run as a service. This applies to all programs spawned by the service directly or spawned by a sub-process of the service, e.g. an external authentication program (nwauth.exe etc.)or a dslave program.

So you must stop the dwatch service in the Services dialog of the Control Panel and it will stop all the dmail servers. If one of the servers or a program that it has spawned still appears on the task list after shutting down the dwatch process then you will have to re-start the machine to make it stop.

On UNIX based platforms:
Simply kill the offending process.

Starting a Server:

Windows Platforms:
If you have stopped just the server then you should start it again by clicking the 'Start Server' button on the appropriate server tab with the DMAdmin administration tool.
If you have stopped DWatch then you should start it again either in the Services dialog of Control Panel or by running the executable (e.g. \dmail\dwatch\dwatch.exe) from a command line. Once started DWatch will start up all DMail servers not already running.

UNIX based Platforms:
Use the startup scripts to start the servers individually or the dwatch process which will start and look after the individual servers. Here are the default locations of the startup scripts:

DWatch: /usr/local/dmail/dwatch/dw_start.sh
or
DSMTP: /usr/local/dmail/dm_start.sh
DPOP: /usr/local/dmail/dpop_start.sh
DList: /usr/local/dmail/dlist/dl_start.sh

Servers Starting at Startup

Windows Platforms:
The DMSetup program adds the dwatch service to the list of services on Windows NT machines. So the DMail servers should start at startup. You can edit the dwatch startup options in the Services dialog of Control Panel.

On non-NT windows platforms the DMSetup installation utility will add the registry key "DWatch" to HKEY_LOCAL_MACHINE - SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\RunServices

You can also simply add a shortcut for the dwatch.exe executable to your startup menu. This option can be good for testing as you can easily kill any subprocesses from the task manager.

UNIX based Platforms:
The DMSetup installation utility should have added lines to your startup script to start the dwatch process at startup, unless you instructed it not to start the servers.

It will generally create a dmail.init file (i.e. /etc/rc.d/init.d/dmail.init) and appropriate links to it for the different run levels. If it hasn't been able to do this then it tries to add startup lines to the file, /etc/rc.d/rc.local

If DMSetup has put them in the wrong place or cannot find the startup script, then you should add the following two lines to your system's equivalent of the file, /etc/rc.d/rc.local

rm -f /usr/local/dmail/dwatch/*.pid
/usr/local/dmail/dwatch/dw_start.sh

Notes:

1. This does not work for the MACOSX platform. The startup scripts are in /etc/startup and you need to edit the mail script, e.g. 1800_mail to look like this,

   *****File Content Within Stars *****
   #!/bin/sh

```
. /etc/rc.common

##
# Start mail server
##

/usr/local/dmail/dwatch/dw_start.sh
**********************************
```

2. The startup scripts set a limit on core file dumps, so that if the program dies a core file is created. for most systems,
   ulimit -c 20000
   sets this. However on platforms like BSDI you need to check that the script files use,
   limit coredumpsize 20000
   or the equivalent for your platform.
3. If your system does not have either of the files,
   /etc/rc.d/init.d/dmail.init
   or,
   /etc/rc.d/rc.local
   then dmsetup will not have put the startup script lines anywhere else.
4. If you are having trouble locating where to put the startup lines then search for where sendmail's startup scripts are.

   For example: The file
   /etc/rc.d/init.d/sendmail
   is often sendmail's startup script, which looks like this:

```
*****File Content Within Stars *****
!/bin/sh
#
# sendmail This shell script takes care of starting and stopping
# sendmail.
#
# chkconfig: 2345 80 30
# description: Sendmail is a Mail Transport Agent, which is the program \
# that moves mail from one machine to another.
# processname: sendmail
# config: /etc/sendmail.cf
# pidfile: /var/run/sendmail.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network
```

```
# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0
[ -f /usr/sbin/sendmail ] || exit 0

# See how we were called.
case "$1" in
  start)
    # Start daemons.
    echo -n "Starting sendmail: "
    daemon /usr/sbin/sendmail -bd -q1h
    echo
    touch /var/lock/subsys/sendmail
    ;;
  stop)
    # Stop daemons.
    echo -n "Shutting down sendmail: "
    killproc sendmail
    echo
    rm -f /var/lock/subsys/sendmail
    ;;
  restart)
    $0 stop
    $0 start
    ;;
  status)
    status sendmail
    ;;
  *)
    echo "Usage: sendmail {start|stop|restart|status}"
    exit 1
  esac

exit 0
**********************************
```

If you locate that file then you can rename it to something else and replace it with this content so that dsmtp is started instead of sendmail (NB: you must replace the paths in this file with the actual paths on your machine if you have not used the default DMail path values):

```
*****File Content Within Stars *****
#!/bin/sh
#
# Startup / shutdown script for dmail servers
#
#
# Start or stop ?
```

```
case "$1" in
  start)
    # start servers
    echo -n "Starting dmail servers: "
    echo
    rm -f /usr/local/dmail/dwatch/*.pid
    /usr/local/dmail/dwatch/dwatch
    ;;
  stop)
    # stop servers
    echo -n "Stoping dmail servers: "
    echo
    tellpop shutdown 10
    tellsmtp shutdown
    cp /usr/local/dmail/dwatch/dlist.pid /usr/local/dmail/dwatch/dlist.exit
    ;;
  *)
  echo -n "Usage: dmail {start|stop}"
    echo
    exit 1
  esac

exit 0
**********************************
```

5. If your operating system has other paths then please let us know and we will list them here:

   FreeBSD is thought to use /usr/local/etc/rc.d instead of /etc/rc.d

---

The Tellpop Command Line Utility

(this page is still under construction !)

Tellpop provides a simple universal command line utility for controlling and monitoring the DPOP system. Of special note is the 'reload' command, which makes DPOP reload its configuration file. It is often used after manually editing the config file, so that you do not have to restart DPOP for the changes made to take effect.

It is used as follows:

Tellpop command parameters

Examples:

Tellpop status	to check status of running DPOP system
Tellpop key 1234-567a-4321-1111	to load new registration key

Tellpop needs access to the same configuration file that DPOP is using. The default location for the configuration file, the default filename being dmail.conf, is system dependent

- `for most Unix variations`
  `/etc/`
- `for Windows 95 and NT`
  `C:\winnt\system32\`

In normal use Tellpop will find the configuration file automatically. When Tellpop is being used to control DPOP running on another machine the configuration file to be used is specified as follows, (example sends the status command).

Tellpop -i another/path/fred.conf status

NB: Tellpop will try to connect to the server domain, as set in your dmail.conf file with the dpop_host setting. If you have not set one then it will do a gethostname( ) type call to find the server's domain name.

Access to Tellpop manager commands can be limited in several ways:

- Firstly a configuration file wildcard list limits the from address for Tellpop manager commands. The limits can be based on ip-name or ip-number wildcard lists, see the manager_ip_address setting.
- Secondly a manager command password is used. This is normally generated on installation and stored in encrypted form in tellpass.dat, see the manager_password setting.
- If Tellpop is being used remotely the manager password command is not normally sent across the network .

Tellpop command summary:

Note: Underscore characters in commands are optional so list_users or listusers can be used.

add_user	Add a username bill to DPOP's INTERNAL list of users.
abort Fred	Abort current connection from user Fred
abort_chan n	Abort current connection on channel n
bins Fred	Shows status of bin files belonging to Fred
bulletin	Create a new bulletin message for all users.
config	List current values of all DPOP's configuration settings
del_user	remove username from DPOP's internal list of users
drop Fred	Converts all messages in Fred's bin files back into a drop file and deletes bin files
drop_all	Converts all users bin files back into drop files. This may take some time.
die	Kills DPOP suddenly and horribly, (i.e. emulation of behavior of some other packages :-)
disable Fred	Temporarily prevent user Fred from connecting to DPOP
enable Fred	Allow disabled user Fred to connect again.
flush_stats	Flush per user stats records to disk. To allow stats command to use current file.
key xyz	Loads a new registration key xyz
list_current	List currently connected users
list_disabled	List currently disabled users
list_users	Provides a list of all users of DPOP who have read mail or been added via add_user command.
log_level	Set the logging level to control what information is stored in the DPOP log file.
offline	Temporarily disables user connections to DPOP (i.e. stops people reading their mail)
online	Enables connections to DPOP. (Lets people start reading their mail again)
password	Saves a new manager password for DPOP. It is stored encrypted in tellpass.dat
prof_init	Initialize counts and averages of profile information
profile	Prints a summary of profile information for DPOP, Gives calls, elapsed and used CPU for key parts of DPOP
register	Presents questions to allow DPOP to be registered. A register.txt file is created for emailing to Netwin.
reload	Reloads the configuration file to ensure any changed settings are put into effect.
sleep n	Puts DPOP to sleep for n seconds.
status	Provides feedback on the current status of the DPOP server. The information returned includes:

shutdown n	Shuts down the DPOP system when current connections = zero or in n seconds
stats	Produces per user connection statistics from .stats files for accounting etc.
statistics	Produces DSMTP usage statistics from the log files.
testfiles	Determines the maximum number of files which can still be opened - can take several minutes on a slow system

---

Main Tellpop Commands

add_user bill

Add a username bill to DPOP's INTERNAL list of users. The main purpose for this internal list is for license key limits.

NB: This does not control access to DPOP. Access can be limited by a variety of different means. See the valid_users setting for information on controlling access.

NB: This does not add email accounts to your POP server. To add users to your email system, see the adding users section.

---

bulletin

Creates a bulletin message for all users. The message is written to a file nnn.txt in the bulletins directory. nnn is the next free bulletin number. The message will be delivered to each user when they check their email.

---

del_user

Del_user fred removes username fred from DPOP's internal user list. Note; if a user is deleted they can still connect to DPOP as the username will automatically be added on connection, however information such as the highest bulletin number that user has read will have been forgotten. User access can be restricted by use of controls on valid username wildcards, valid from ip name wildcards, valid Unix usernames and by disabling individual users.

---

password

The command Tellpop password xyz will set a new manager password for DPOP. An encrypted form of this password is stored in the tellpass.dat file and is used automatically by DPOP, Tellpop and DMAdmin. The password is normally encrypted

with a random token whenever it is sent across a network but can be used unencrypted if DPOP is being controlled manually using Telnet.

---

status

Provides feedback on the current status of the DPOP server. The information returned includes:

- DPOP version number.
- The time DPOP was started and current up time.
- The number of current connections, the peak number of connections and the maximum number of connections allowed
- The current license class and user limit

---

stats *jul*

Provides per user connections statistics for all users from matching statistics files. DPOP produces one .stats file per day with names dpopmmmdd.stats so the example above would process all stats files for July. This can be used to provide accounting or charging information. For each user it lists:

- Username
- IP number for last connection from this user
- Total and average connection time in seconds and average connection time for checks when no mail was available.
- Total number of connections and average time between connections
- Total messages transferred and total quantity in bytes

A summary at the end gives grand totals for all users.

The output is written to the file, dpop.sum

---

flush_stats

If you want to use the Tellpop stats command to lookup statistics for the current day you should first issue the flush_stats command to make sure current records are written to file. Otherwise the last few connections will not be included in the summaries.

---

register

Presents questions to allow DPOP to be registered. A register.txt file is

created for emailing to Netwin.

Payment details are encrypted for safe transfer to Netwin Ltd..

---

key xyz

Loads a new registration key xyz

---

offline

Puts DPOP in offline mode in which new connections from users are disabled. Generally used prior to shutdown to ensure no current sessions are aborted. Can also be used to temporarily disable connections without stopping DPOP. Naturally DPOP will still accept manager command connections.

---

online

Puts DPOP back in online mode which enables normal client connections.

---

shutdown

Shuts down the DPOP system.

---

statistics

Produces DSMTP usage statistics from the log files.

NB: Requires that the DSMTP setting, oneline_stats is set to true in dmail.conf to create the dmDDMM.log files used. This command was added in version 2.9a

For each domain or user the number of messages sent, received, transfered or failed are displayed plus the respective total number of bytes associated with these messages. For each value the peak and average values are given and the time band in which the peak value occured.

Usage:
statistics [-U|D][-R|-S][-Dn][-Pn][-An][-sortby] timeframe

-D Calculates statistics by domain name
-U Calculates statistics by user name

-S Calculates statistics by sender or senders domain
-R Calculates statistics by receiver or receivers domain
-Dn Specifies the top n number of domains or users to display
-dall will display all users or domains
-d0 will display none(requires significantly less system resources)
-Pn Set size of the peak window(n) in seconds
-An Set size of dominator(n) used to calculate averages (in seconds)
-sortby Indicates statistic by which domains or users are to be compared.
Options: recv|sent|delv|fail|brecv|bsent|bdelv|bfail
timeframe Period over which analysis will be preformed.
Options: [MM/dd] [hh.mm.ss] to [MM/dd] [hh.mm.ss]

hour	the last hour
day	the last day
week	the last week
month	the last month
so_far_today	statistics since midnight

Examples: tellpop statistics 1/15 to 1/20
Analyse period from the start of the 15th Jan to the end of 19th Jan
tellpop statistics -d100 -u hour
Displays the 100 users who logged the most sends in the last hour tellpop
statistics -p3600 -a1800 month
Analyse last month giving the peak hour(3600s) of usage for each given statistic. All averages will be quoted per half hour(1800s).
Defaults: statistics -d -s -d10 -p600 -a600 -sent so_far_today

---

reload

Reloads the configuration file to ensure any changed settings are put into effect.

NOTE : If you have modified a dmail.conf setting which is relevant to both DPOP and DSMTP, it is necessary to reload **both** servers individually with their reload configuration file commands, e.g. tellsmtp reload **and** tellpop reload. DMAdmin will do this automatically, but note that you can send the reload command from the command list in DMAdmin whenever you wish.

If you have an exceptionally large dmail.conf file this process of reloading could take as long as 30 seconds. You should not need to do reloads of the configuration file for DPOP very often so this sort of down

time should not be too much of a problem. If it is then please contact support-dmail@netwinsite.com to discuss your situation.

---

list_users

Provides a list of all users of DPOP who have read mail or been added via add_user command.

---

log_level

This command will change DPOP's log level during runtime. It will not cause DPOP's config file to be re-written, so next time DPOP is run, it will use the log level setting found there. This command is useful for closely observing a particular transaction DPOP may be about to make.

The available log levels are:

```
error:  the only information written will be
        errors, warnings, socket read and
        write information and minimal
        progress information.

info:   as well as the error information,
        this setting gives much more
        progress information, as well as
        file open and close calls.

debug:  as well as the info information,
        this setting gives a whole lot of
        internal status information,
        function calls...all sorts of stuff.
        However, it may slow down operation
        slightly and can produce large log
        files, so use sparingly.
```

If DPOP is crashing or doing very peculiar things, the most useful things to Netwin support are the config file and the log file.

Info is the default setting, and the usual setting for operating in. For more information see the **Logging of information and error messages** section.

Additional Tellpop commands for testing

drop_all

(Normally only used before reverting to another POP3 server and when 'read but undeleted' messages are left)

Turns all bins back into drop files for all users. Connection to DPOP while this command is executing is disabled.

Note: This command deletes the bin files for each user after successful conversion.

---

drop uuu

Turns user uuu's bin files back into a drop file.

Note: This command deletes the bin files if conversion is successful.

---

bins uuu

(Normally only used by Netwin - documentation only provided for completeness.)

show status of user uuu's bin files in terms of message numbers in each bin

Columns are bin number, file size, used bytes, number of messages in bin, (message numbers)

---

xtest n

(Normally only used by Netwin - documentation only provided for completeness.)

Add n users with names test0 up to testn

---

xusers n

(Normally only used by Netwin - documentation only provided for completeness.)

Add n users with random names

---

testfiles

(Normally only used by Netwin - documentation only provided for completeness.)

Check how many files can be opened concurrently. i.e. check number of free file handles available. Note: on average two handles are needed for each concurrent client connection to DPOP. Concurrent connections on

some platforms is limited by the availability of file handles or descriptors.

---

die

(Dangerous! Normally only used by Netwin - documentation only provided for completeness.)

This command simulates the behavior of certain other much loved software i.e. Causes DPOP to perform an illegal command and die, generally producing a core dump etc.

Back to Top of Summary List

Products Downloads Prices Support Company

DMAdmin (DMail Manager)

Under Construction :-)

DMAdmin is a GUI windows administration tool for the DMail server. It is intended for use by the system administrator. You can use its remote administration option to administer the DMail server running on any platform from a windows machine.

We are in the process of modifying DMAdmin so that it is more up to date with the full capabilities of the DMail server. This section will not be completed until that has happened. If you are unsure of anything in DMAdmin then please remember that you can edit the dmail.conf file directly, DMail Support will also be happy to help.

DMAdmin Contents:

Overview

. . .

Main Window

. . . **Config Button**

. . . **Start All**

. . . **Stop All**

. . . **Users**

. . . **Exit Button**

. . .

Configuration Tabs

. . .

DMAdmin DMAdmin is a graphical user interface for controlling DSMTP, DPOP, DList and their configuration settings. This is available for Windows 95 and Windows NT but can also be used to control Unix versions of DMail/DPOP remotely.

Unix installation

Copy dmadmin.exe and dmail.hlp to your PC using a 'binary' transfer mode, install it in a dmail sub directory and then add it to your button bar in the usual way. The default directory should be set to c:\dmail or wherever you installed it on your PC.

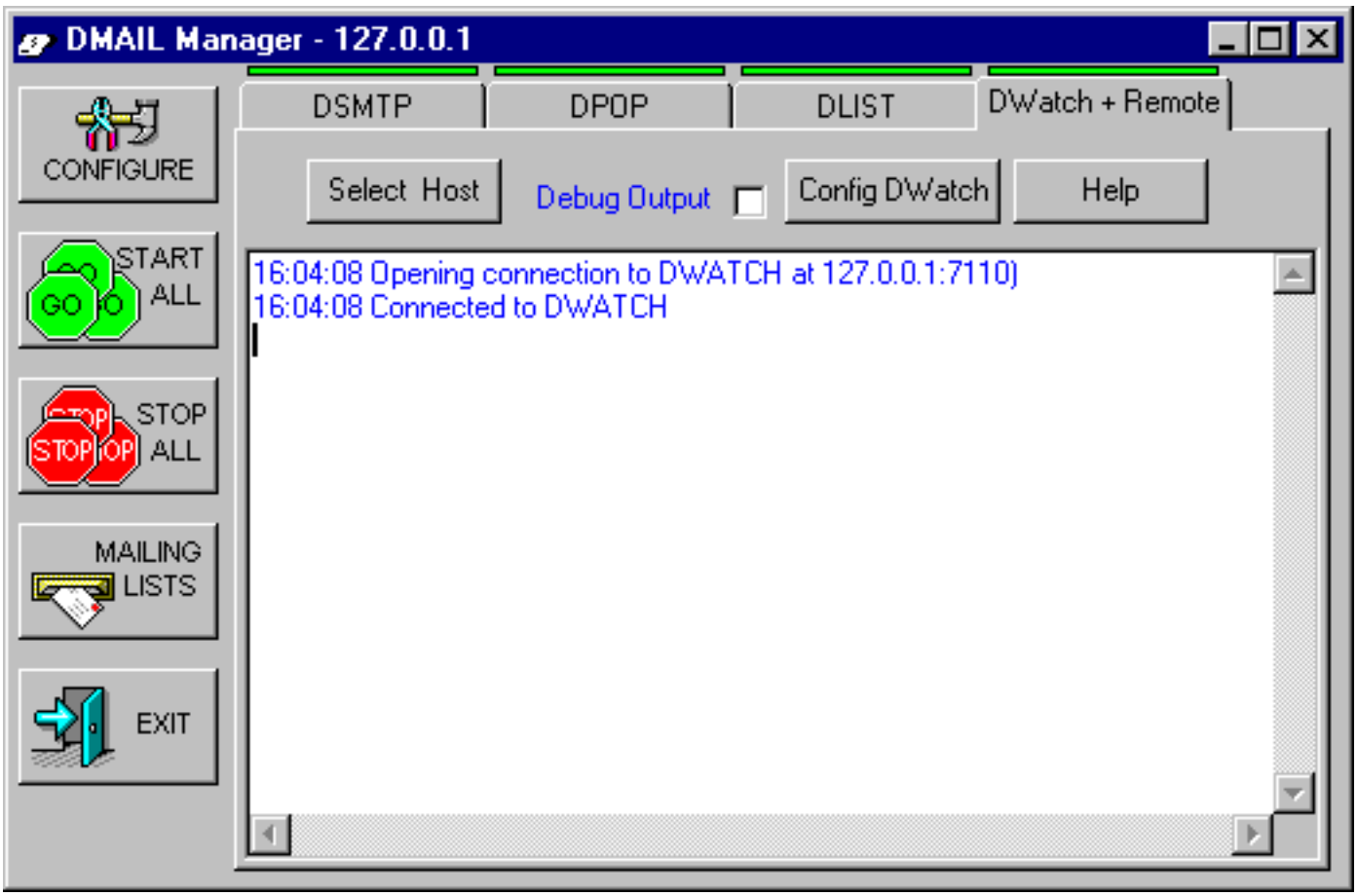
Using DMAdmin remotely

First you must set the DMail administration password, to do this type in on the actual machine running DMail:

```
tellpop password xxxxxx (replace xxxxxx with a password)
```

Then in DMAdmin you can select your mail server as the host to administer and type in the same password, then it will be able to control your mail server remotely.

Screen Shot



Products Downloads Prices Support Company

IMAPD Server

If you have users who wish to connect to an IMAP server instead of a POP server then you can install the IMAPD server alongside DPOP.

Windows NT Installation

Download the latest version from the Utilities Download Page
These instructions assume that you have installed dmail version 2.5h or later in the directory DMAIL_DIR.

- Unzip the imapd package to a temporary directory.
- >From the temporary directory, type
  ```
  copy imapd.exe DMAIL_DIR
  copy imapdsvc.exe DMAIL_DIR
  copy DMAIL_DIR\dwatch\dpop.wat DMAIL_DIR\dwatch\imapdsvc.wat
  ```
- Edit DMAIL_DIR\dwatch\imapdsvc.wat and change the 3 occurrences of dpop to imapdsvc
- Restart DWatch:
  To do this:
  * Go to the control panel.
  * Double click on services.
  * Select "dwatch monitor for dmail servers".
  * Click Stop.
  * Click Start.
- Imapd should start working in a few seconds.
- To check that Imapd is working correctly, telnet to port 143 and you should get a one line introduction message. eg:
  ```
  telnet 127.0.0.1 143
  ```
  should say something like
  ```
  * OK dev.netwin.co.nz IMAP4rev1 v4.4.3k (Mar 8 2000) server ready
  ```
  To terminate the imapd connection type:
  ```
  a logout
  ```
- If imapd is not working correctly, instead try to run it from the command line by typing:
  ```
  DMAIL_DIR\imapd
  ```

If it still doesn't give the introduction message, it should give a reason, or alternatively take a look in DMAIL_DIR\log\imapd.log for a possible reason.
If it works correctly from the command line, but not through a telnet session, you have probably incorrectly set up imapdsvc or incorrectly configured it for use with DWatch.
- Check that imapdsvc is running, and if so, take a look in DMAIL_DIR\log\imapdsvc.log for a possible reason why it is failing.
- If imapdsvc is not running, take a look at DMAIL_DIR\log\dwatch.log for a possible reason why it is not starting imapdsvc.

Unix Installation

Download the latest version from the Utilities Download Page
These instructions assume that you have installed dmail version 2.5h or later in the directory /usr/local/dmail.

- Unzip the imapd package to a temporary directory.
- Ensure that imapd is owned by root, and (for versions 4.4.3f or later) that imapd has 6755 permission. These can be accomplished using
```
chown root:root imapd
chmod 6755 imapd
```
- From the temporary directory, type
```
cp imapd /usr/local/dmail
```
- Edit /etc/inetd.conf
  - Find the existing imap line, which is the line beginning with the word **imap**. For example on linux, the line will look something like
```
imap    stream     tcp     nowait       root
/usr/sbin/tcpd     imapd
```
  - Change the line to specify the location of the dmail imapd server.
    For example on linux, you might change it to:
```
imap    stream     tcp     nowait.100   root
/usr/sbin/tcpd     /usr/local/dmail/imapd
```
    Or on Solaris, it might be:
```
imap    stream     tcp     nowait.100   root
/usr/local/dmail/imapd     imapd
```
  - If there is no existing imap line, you must create one using the above syntax, or using syntax similar to other lines in the file. The exact syntax may vary between different platforms.
  - The .100 value after the nowait parameter specifies that a maximum of 100 connections to imapd can occur within 1 minute. This value defaults to 40 on linux if omitted. You may want to raise or lower this depending on your expected number of users. If this limit is exceeded, inetd will refuse any connections for the next 10 minutes before permitting them again.
- Force inetd to reload.
  On most linux systems this can be accomplish with:

```
killall -HUP inetd
```

On other systems you could find the process ID of inetd (using `ps x | grep inetd`) and try:

```
kill -s HUP [process ID]
```

or something similar.

- Imapd should now be working
- To check that Imapd is working correctly, telnet to port 143 and you should get a one line introduction message. eg:
  `telnet 127.0.0.1 143`
  should say something like
  `* OK dev.netwinsite.com IMAP4rev1 v4.4.3k (Mar 8 2000) server ready`
  To terminate the imapd connection type:
  `a logout`
- If imapd is not working correctly, instead try to run it from the command line as root by typing:
  `/usr/local/dmail/imapd`
  If it still doesn't give the introduction message, it should give a reason, or alternatively take a look in /usr/local/dmail/log/imapd.log for a possible reason.
  If it works correctly from the command line, but not through a telnet session, you have probably incorrectly set up inetd. Possible reasons include multiple imap lines in inetd.conf, wrong path specifications or failing to restart inetd properly.

Configuration

IMAPD currently supports only some of the features of DPOP/DSMTP. These are any features which use the following DPOP/DSMTP settings which appear in dmail.conf.

authent_domain	true for 'user@domain' instead of 'user' to be passed to the authentication process (external authentication only)
authent_method	sets the method used for user/password lookups
authent_process	Specifies the executable to use for external authentication process(when authent_method external)
authent_timeout	Timeout (in seconds) for external authentication requests.
bulletin_path	Directory to contain bulletin messages of form nnn.txt.(IMAPD version 4.4.3i or later)
bulletin_from	Text to be sent as from line in all bulletins. Default is 'Email System Administrator'.(IMAPD version 4.4.3i or later)
dpop_host	The TCP/IP address of the host DPOP/IMAPD is running on.
drop_path	specifies the directory to use for email drop files
drop_prefix	If true then the virtual domain prefix is used as part of path for drop files.

forward_user	Anti-SPAM; turns on an anti-relaying system that allows relaying only after a recent DPOP/IMAPD login.
forward_window	Anti-SPAM; sets time (in seconds) in which relaying is allowed after POP/IMAPD login (i.e. expiry time of forward_user records).
hash_spool	Sets hashing method (0,1 or 2). Hashing is where drop_files are distributed across multiple directories.
host_domain	adds a domain name to the list of domains to be recognized as being local
lock_id	(UNIX); sets a file locking id for multi-server systems.(IMAPD version 4.4.3e or later)
log_level	Specifies how much information to output to the logfile (one of error, info, debug).
log_path	Specifies where the server log files are to go.
lowercase_username	Now a common setting to both DSMTP,IMAPD, and DPOP (2.4e and above).
lowercase_password	If true DPOP/IMAPD will always set passwords to lower case before using them for authentication (auth module must have lower case passwords or be case insensitive).
maildir_home	Specifies path for all incoming maildir files as well as created imap folders (IMAPD version 4.4.3m or later)
maildir_althome	Alternative maildir path for a virtual domain (eg maildir_althome vdomain1.somewhere.com /shared/spool/maildir/vdomain1) (IMAPD version 4.4.3m or later)
max_log_size	Max size, in bytes, for DPOP/IMAPD log file before renaming and starting new one.
user_ip_address	IP addresses user connections can come from.
user_quota	Enables a per-user mailbox quota system and optionally sets the default quota(true/false/kbytes).(IMAPD version 4.4.3g or later)
use_maildir	Use maildir for storing of inbox and all imap folders. Must also specify maildir_home (IMAPD version 4.4.3m or later)
valid_users	Valid usernames to get their mail here.
vdomain	Sets up a virtual domain for use with/by DPOP, DSMTP, and IMAPD.
vdomain_passwd	Tells DSMTP, DPOP, and IMAPD to use a separate passwd file for a vdomain (Unix)
vdomain_separator	Specifies the separator character to use with Virtual Domain prefixes.

IMAPD adds two additional dmail.conf settings. Most users need not use these settings.

imapd_mailbox_name

By default, all imap mailbox directories are created in the drop path, with the same name as the dropfile, but with .mbx appended. Only specify this setting if you want all standard unix users (including virtual domain users which use the standard password file) to have their imap mailboxes stored in their home directory, in which case, they

will be stored under this name. This may be useful if your users also use pine, in which by default, all folders created are stored in the directory ~/mail.

```
Example:
imapd_mailbox_name mail
```

imapd_authent_process

Specifies an alternative authentication process to use instead of the standard authent_process value. If not specified, imapd uses the value of authent_process instead. (IMAPD version 4.4.3s or later)

```
Example:
imapd_authent_process /usr/local/dmail/nwauth
```

Notes:

1. When using user_quota, keep in mind that if users use both dpop and imapd, then they will receive this quota for both programs, so can effectively have twice their normal quota. Additionally, imapd will allow users to go slightly over quota if they were under quota before attempting to copy/create a message or folder.
2. Reading mail with both dpop and imapd is not advisable, since messages read with one can not be read with the other.
3. Imapd needs to read your config file "dmail.conf" for every new connection. By default it ignores #include lines as it could take too long to read them all. You can force imapd to follow these using the setting "imapd_include_level". This defaults to 0, and specifies how many #include levels to follow. For example you may add

   ```
   imapd_include_level 1
   ```

   to the start of dmail.conf to tell imapd to follow #include's specified in dmail.conf, but ignore #include's within the included files. Following #include's is only supported in imapd 4.3.3q or later.

Checking it works

If you have successfully set up imapd so that you can successfully telnet to it as described in the installation instructions, then you can now try and see if user authentication and message reading is working. First send a message to the user, but don't read it using any pop/imap server. Then telnet to your imapd server:

```
telnet 127.0.0.1 143
```

Imapd should respond like:

```
* OK dev.netwin.co.nz IMAP4rev1 v4.4.3k (Mar 8 2000) server ready
```

Type:

```
a login username password
```

where you replace *username* and *password* with the appropriate values. Imapd should respond like:

```
a OK LOGIN completed
```

Then type:

```
b select inbox
```

Imapd should respond like:

```
* 1 EXISTS
* 1 RECENT
* OK [UIDVALIDITY 964139896] UID validity status
* OK [UIDNEXT 2] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)]
Permanent flags
* OK [UNSEEN 1] 1 is first unseen message in
/shared/spool/mail/g/test
b OK [READ-WRITE] SELECT completed
```

Then type:

```
c uid fetch 1 body.peek[]
```

And Imapd should respond like:

```
* 1 FETCH (UID 1 BODY[] {317}
Received: from matt ([10.0.0.24]) by dev.netwin.co.nz ; Wed, 16 Aug
2000 14:11:15 +1200
To: test@dev.netwin.co.nz
Subject: Hello Test
From: test2 sender
Message-ID: <96639187501@dev.netwin.co.nz>
Date: Wed, 16 Aug 2000 14:11:15 +1200
X-Rcpt-To:

line 1 of 4
line 2
line 3
line 4
)
c OK UID FETCH completed
```

Finally type:

```
d logout
```

To end terminate the connection.

If after sending the login information, imapd responded:

```
a NO LOGIN failed
```

then first check that you can login to dpop using this login. eg:

```
telnet 127.0.0.1 110
+OK DPOP Version 2.8q. <343.966394324@localhost>
user username
+OK test nice to hear from you - password required
pass password
+OK burst ok test has 1 msgs
quit
```

Where *username* and *password* are replaced by a valid username and password.

If that does not work, then the problem is with your dpop setup - refer to dpop documentation to fix the problem.

If that works, then something is probably wrong with your imapd setup. Try setting the dmail.conf log_level setting to debug, perform the test again, and have a look at the end of the imapd.log file. (in the log sub-directory in your dmail directory. e.g. /usr/local/dmail/log/imapd.log or c:\dmail\log\imapd.log) This might mention an obvious problem that you can fix. If not, send the imapd.log file together with your dmail.conf file to us at support-dmail@netwinsite.com and we might be able to help.

Downloading the DMail Server Package

DMail is a product of Netwin Ltd. You may download and use this software for 1 month. At the end of this evaluation period you must register your copy of the software and purchase a software license or stop using it.

Select a version to download from the list of supported operating systems below.

If you would like to run DMail on a system not in the current list please let us know by sending mail to support-dmail@netwinsite.com.

If during the evaluation period you require technical support or have questions about DMail please send email to support-dmail@netwinsite.com and we will try to help you.

NB: Linux users you have to choose between, **linux_libc6**,for the newer version 6 C libraries in distributions like Red Hat 5.0 and above, and, **linux**, for the older version 5 C libraries. If you have the file, /lib/libc.so.6 then you should use the libc6 version.

Operating System	DMail (dsmtp,dpop,dlist)	File Size
Linux (libc6)	dm27u_linux_libc6.tar.Z	3,800 K
Linux (old pre RH5.0)	dm27u_linux.tar.Z	3,675 K
Windows 95/98	dm27u.exe	3,382 K
Windows NT	dm27u.exe	3,382 K
Digital Unix(OSF)	Building, contact us	2,894 K
Free BSD	dm27u_freebsd.tar.Z	2,048 K
HP-UX	Building, contact us	3,464 K
MacOSX	dm27u_macosx.tar.Z	2,115 K
Solaris(sparc)	dm27u_solarissparc.tar.Z	4,157 K
Solaris(x86)	Building, contact us	3,389 K
BSDI 3	Building, contact us	3,808 K
BSDI 4	Building, contact us	3,559 K
AIX	Building, contact us	2,552 K
Linux_MIPS (incl. Cobalt)	Building, contact us	- K

SCO	Building, contact us	- K

The full distribution sets above contain the three server applications DSMTP, DPOP and DList, a setup wizard DMsetup, Utilities, Manuals and Examples.

Please read our License agreement, License.htm, which applies to all of the software downloadable from this page.

Installing DPOP by itself

You can install the POP server, DPOP by itself. Download the normal DMAIL distribution and then run ./dpopsetup instead of ./dmsetup (this only applies to Unix)

For existing users . . .

DMail Utilities

Netwin and our customers have produced Utilities for use with DMail. You can download these from this page.
DMail Utilities Download Page (Including IMAPD, ODBC Authenticator, Columns ...)

Beta Directory

New versions of the software will normally be made available in the beta directory first. Software in this directory is still undergoing testing, so download and use it entirely at your own risk. We will of course be very interested to know of any bugs you find in the beta versions. The beta directory for DMail is ftp.netwinsite.com/pub/netwinsite/dmail/beta

You should check the updates page for details of the beta versions.

Bulletin Facility

In addition to mailing lists DMail provides a bulletin facility. This is useful when you need to send an email message to all users but without the overheads of duplicating and sending the same message to all users. The bulletins are organized as follows:

- A bulletin directory is specified in the dmail.conf file with a setting: e.g. bulletin_path /var/mail/bulletins
- This directory contains files with names of the form nnn.txt, where nnn is the bulletin id number
- Any user connecting to DPOP will receive any of these files they have not already seen as an email message
- For each user DPOP stores the the id number of the last bulletin seen by this user.

To produce a new bulletin it is necessary to create a new file containing the bulletin, with a name nnn.txt and place it in the bulletin directory. The bulletin id number needs to be greater than the id for any bulletins already used. The tellpop bulletin command can be used as a quick way of producing a new bulletin thus:

C:\>tellpop bull

Tellpop will then respond with:

Tellpop 2.00
Creating bulletin 5. Enter text finish with .<cr>

So then you can enter your message (remember to finish with a full stop on a line by itself):

>The system will be down due to a <cr>
>power cut from 2 to 5pm on Friday <cr>
>we hope this will not inconvenience users <cr>
>John Smith <cr>
>System Administrator <cr> >. <cr>

Old bulletins can be left in the directory or deleted periodically. Any user which has not seen a bulletin when it is deleted will not see it at all. In addition to prevent bulletin id numbers increasing for ever the following scheme is used: When DPOP starts up if the bulletin directory is empty then all users will have their "last bulletin read id" set to zero. Bulletins id's can then start at one again.

Using the bulletin facility is in some ways similar to sending an email to a mailing list with everyone on it. However it is much more efficient. For example if you have a system with 20,000 users sending an email to all users involves considerable processing and twenty thousand copies of the message are stored on disk, one for each user. In contrast using a bulletin on the same system completely avoids the processing involved in sending the emails to a list. It also saves only one copy of the bulletin. When each user checks for email they then obtain their own copy of the bulletin which, in general they will delete from the server immediately. However, if they do choose to leave it on the server they then have their own copy just as though it was an ordinary email message.

The message headers for a bulletin message are set as follows:

From:	Set by bulletin_from setting in dmail.conf Default value is "Email System Administrator"
Subject:	Bulletin nnn

NB: Configurable subject lines for bulletins will be coming soon (written 19 August 1999).

User Administration

To add users to your email system . . .

The DMail servers can use one of a number of systems for user authentication. Unfortunately this choice can make selecting an authentication system more confusing than it needs to be.

NOTE: by default the DMail servers use your system user database. So if you have just installed DMail try sending a message to a local user on your machine.

The choices for authentication:

You choose the authentication method with the configuration file,dmail.conf, setting, authent_method which can be one of:

1. **authent_method nt_user**
   which means use the windows NT user database, this is the default on Windows NT machines.

   To add users you should run the User Manager (under Administrative Tools) and add users as if you were adding system accounts. Given that DSMTP and DPOP can autheticate a user, they will create a drop file for that user as necessary.
2. **authent_method unix_user**
   which means use the /etc/passwd file and standard system calls (e.g. getpwnam())and is the default on UNIX based platforms.

   Add users as you normally would to the system, as system users automatically have email accounts on with this option. The DMail servers will create drop files etc. as necessary.
3. **authent_method external**
   which means that the servers should spawn a specified external authentication program which can in turn be talking to any kind of database. External authentication programs are the link between the dmail servers and any type of user database you wish to use.

The next section covers external authentication extensively, but please note the following things which apply to any authentication method. (You should probably skip these for now but come back to them once you have read the more general material.)

1. After changing any authentication setting you MUST stop and re-start the DSMTP and DPOP servers
2. For your main (first)domain you can probably add simple usernames. Once you add any virtual domains you will need to decide how to re-use usernames on the virtual domains. See Username Re-Use in the sections below.
3. If you want users to be able to add themselves automatically, then see Setting Up a Web Based Email system with Auto Account Creation
4. Notes on Case Sensitivity for all Platforms:

5. Notes on UNIX Authentication
6. Notes on NT Authentication

External Authentication of Users

DPOP and DSMTP allow the use of an external application for authenticating client connections. The servers run (spawn) multiple copies of the authentication application.

In this section:

1. Choosing an external authentication module to run
2. Telling DMail to use your chosen external authentication module
3. Adding users to an external authentication module
4. Creating your own external authentication module
5. The External Authentication Protocol

1. External Authentication Modules List

You will have already downloaded some external authentication modules with DMail. If you wish you can write your own external authentication program (we suggest you modify nwauth).

The following is a list of external authentication programs available from us. The links take you to detailed information.

Ext. Auth. Program (click for details)	Download From	Comments
NWAuth	In every DMail distribution set	This is our recommendation. Source provided. Please feel free to use it as a base if you are writing your own module.
UNIXAuth	In 2.7 DMail distribution set	This is our module to talk to the UNIX system password file.
NTAuth	In 2.7 DMail distribution set	This is our module to talk to the NT system user database.
LDAPAuth	Utilities Download Page	authenticate with LDAP server
DNAuth	Utilities Download Page	A version of nwauth which can check and lookup users from a DNEWS users.dat file.
MySQLAuth	In 2.8e and above DMail distribution set	Our own authentication module for talking to a MySQL server.

ODBCAuth	Utilities Download Page	Our own authentication module for talking to an ODBC Driver for a Database (e.g. MS Access, MS SQL Server, ORACLE)

The following is a list of external authentication programs offered by our customers.

Ext. Auth. Program	Located	Comments
MYSQLauth	**Replaced by our module above.** For details email support-dmail@netwinsite.com	Source of an authentication module which talks to a MySQL server.
ODBC Authenticator	Utilities Download Page	Free module that talks to ODBC on NT. (source available for small fee)

2. **Telling DMail to use your chosen external authentication module**

   Set the following TWO compulsary settings in the configuration file, dmail.conf,

   authent_method external
   authent_process executable_file

   (where 'executable_file' is the full path and filename of the executable to be run)
   AND then you must STOP and RE-START both DSMTP and DPOP.

   For example, for NWAuth which you will find in the DMAil distribution set,

   authent_method external

   and then

   authent_process \dmail\nwauth.exe
   or
   authent_process /usr/local/dmail/nwauth

   for Windows and UNIX platforms respectively (assuming default installation directories).

   Two recommendations:

   1. If you don't have any limitations on the your authentication database then we recommend that you use NWAuth.
   2. We also recommend the following setting,

      authent_domain true

      which forces user@domain to be passed to the external authentication process.

   Some examples of other settings you might want to set:

authent_number 3	# tells DSMTP/DPOP to run 3 authent processes at once
authent_timeout 20	# sets DPOP's length of time until timeout to 20 seconds
authent_cache 500	# tells DPOP to cache 500 authentication requests

3. **Adding users to your external authentication module**

   The way to add users to your external database is hopefully known to you. However, all external authentication modules should adhere to the protocol so you should either be able to run the authentication program from the command line or use DMAdmin (windows GUI) or NetAuth(web admin) to add users as well.

4. **Creating your own external authentication module**

   External authentication can be used for adding special connection restrictions. For example allowing email from laboratory computers only outside working hours. A sample authentication program is supplied with the distribution set, the C code for it is shown on this page, nwauth.c. On NT use Visual C to build an authent process as pipes don't appear to work with some versions of Borland C.

   You must also provide a path to the executable you want to use. Let us assume you wish to use an application called myauth.exe for authentication. The application myauth.exe must read commands from standard in and write replies to standard out. The commands and replies defined allow the same system to be used by both DPOP and DSMTP. Note some parts of the reply are not needed for DPOP.

   Your authentication application must use Netwin's Simple Authentication Protocol, follow the link in the next section for details, page.

5. **The Defintion of the External Authentication Protocol**

   Follow this link for a full definition and desctiption:
   External Authentication Protocol

Repeating Usernames on Virtual Domains (username re-use)

If you have set up dmail for multiple domains then the servers need some way to differentiate between user bob on one domain and user bob on another domain.

Importantly you don't want the end user to have to use strange usernames because bob on each domain wants to be able to log in with the username 'bob'.

There are three ways in DMail to handle this. Note: these systems may seem complicated, but they have

been worked out so that life is simple for the end user.

1. Don't allow usernames to be repeated on other domains:
   If you are happy with this then you probably don't need any virtual domains. You can specify all your domains as synonyms of the main domain with host_domain settings.
2. Add the users to your database in the form user@domain: **(recommended)**
   To do this you set in the configuration file, dmail.conf,

   authent_domain true

   (remember to restart DSMTP and DPOP after changing an authentication setting)

   When a server needs to authenticate a user 'bob' it checks the external authentication for the user, bob@domain.
3. Add a prefix to the the usernames of virtual domain users:
   Leave your existing 'main domain' users and add all virtual domain users with a prefix on their username, e.g. dom1_bob .

   This system is particularly useful if you already have a database full of usernames on your main domain, but wish to add a virtual domain and hence a new group of users.

   For Example:
   You already have a user, bob. And you want to add another domain, domain2.com. So you add a vdomain line to dmail.conf,

   vdomain dom2 1.2.3.4 domain2.com /mail/domain2

   the second entry sets the vdomain_prefix to 'dom2'. So in the user database you can add a user bob to this second domain by adding the username, dom2_bob. (the setting vdomain_separator can be used to set '_' to something else).

   The user still logs in with the username bob but he connects to the ipaddress 1.2.3.4 so DPOP knows to authenticate the user, dom2_bob.

Controlling client access to DPOP

DPOP provides a variety of methods for limiting access from clients trying to read mail.

First the client must use a valid username. This is limited by a wildcard list of valid names in the configuration file.

There must be a drop file for that user.

The username/password combination must be valid. This can mean a valid Unix or NT user/passwd. A variety of other methods for checking this combination are provided for and controlled by settings in the configuration file. An external authentication routine can also be used: See Adding Users ....

The from address ip_name or number can be limited using wildcard lists in the configuration file

By using an external authentication process other controls may be imposed such as limiting access from certain classes of users or users of certain machines to specific times of day. For example student users on laboratory machines may not be allowed to read mail during work hours. An example external authentication program is provided with DPOP. See the next section for details.

Controlling DSMTP access

Users do not log in to an SMTP server, but in general all mail delivered by an SMTP server has both a source address and a destination address (username and domain). This allows you to restrict mailing based on username and domain. **You can restrict who can send mail out of your smtp server (non-local delivery and relay):**

1. Forward User System:
   The settings, forward_user and forward_window set up a system that allows local users to relay but stops your site being an open relay.
2. ...

You can restrict who can send mail in to your smtp server (local delivery):

1. ban_ip
2. ...

Notes on UNIX Authentication:

- By Default (unix_case false), DSMTP will first do a case sensitive lookup on the username of an incoming message. If that fails then DSMTP will do its own case insensitive 'walk' of the password file, and consider the lookup successful if it finds one, unambiguos match. See Case Sensitivity
- Version 2.5c of DSMTP does the standard getpwnam() call if DSMTP's case insensitive passwd file walk fails. Then if that fails, and unix_case is set to false, DSMTP lowercases the username and does the getpwnam() call on the lowercased form (note that this not full case insensitivity!!).
- The getpwnam() system call will only work with the Yellow Pages system call and the shadow passwords if you use the linux libc6 build version.

Notes on Password Files (UNIX based systems only)

Password files can be specified for individual domains with the dmail.conf setting,

vdomain_passwd <domain> <filename>

The file specified (with full path) has exactly the same format as the /etc/passwd file.

The DMail servers do their own walk through these password files and will use any home directory setting etc. found within.

Notes on Case Sensitivity

On setting up a system you should be careful to get your case requirements and the settings to carry them out correct from the start, because it can be time consuming to fix problems once your system is up and running.

The settings that affect case sensitivity in DMail are,

lowercase_username
lowercase_password
unix_user

The following only applies to versions 2.4e and above.

The default settings are:
lowercase_username false
lowercase_password false
unix_user false (Unix based platforms only)
which means that the following situtations exist:

On Windows Platforms:

The username on an incoming message will be looked up by DSMTP in the case sensitive form. DPOP will also do this by default.

On Windows NT the username lookup is not normally case sensitive, so mail will be accepted by DSMTP for users bob, BOB, BoB etc. so long as there is a user called bob of some sort in the user database. Similarly DPOP will allow users to use any case for their username entered in their email client when they connect to read mail.

Both servers will write or read from the drop file with the case as it is presented to them. On NT this does not cause any problems as the file
\dmail\in\BOB
is the same file as
\dmail\in\bob

NB: The above means that on windows platforms you cannot have separate users, BOB and bob, i.e. differentiate usernames based on case differences in the username.

If you are using your own external authentication module do be aware of the case sensitivity of the lookup. DSMTP cannot currently be made to do a non case sensitive lookup. e.g. if mail arrives at DSMTP for BOB@domainx.com, then DSMTP will send
lookup BOB
to the external authentication module.

Our authentication module, NWAuth.exe always does a case insensitive lookup, so on windows platforms all problems are avoided.

If you want users to be able to enter their password in any case then set the lowercase_password setting to true and make sure that you have created the user with a lowercased version of their password.

On Unix based platforms:

DPOP by default will do a case sensitive lookup of the username. It will also look for a case sensitive drop file. E.g. if the user kathy has in her email client KATHY as her username, then DPOP will lookup the user KATHY and if it gets a match then it will look at the drop file,
/var/mail/KATHY

DSMTP is a little trickier :-)

By default on Unix DSMTP will first do a case sensitive lookup on the username given in the incoming message. If that fails, then it will do its own case insensitive lookup of the password file - this involves DSMTP doing its own password walk through the /etc/passwd file, rather than using the standard Unix calls. If there is a specific password file set, with the setting, vdomain_password, for the domain of the incoming message, then DSMTP will 'walk' that password file instead of /etc/passwd in the same manner.

In versions prior to 2.4j there was a bug in the default behaviour. DSMTP, when it did the case insensitive passwd file walk and found a unique match, would still use the version of the username that it started with for the drop file name, instead of using the drop file name as it found it in the password file. This resulted in drop files of the incorrect case, which DPOP then did not find. Setting lowercase_username to true also fixed this problem, see below.

If lowercase_username is set to true then DPOP will always do a lowercase lookup and will look for a lowercase drop file.

To make DSMTP only do case sensitive lookups you must set the unix_case to true. NB: this means that all incoming mail addresses to incorrect case versions of the username in the password file will be bounced!

Setting lowercase_username to true, will make DSMTP create ALL drop files in lowercase (as it will for DPOP also). NB: It will not change how DSMTP does its username lookups.

So on Unix we recommend:

- If you are using NWAuth or another External Authentication Module that does lowercase

lookups:
Set lowercase username to true, so that the drop files are always in lowercase form. NB: you cannot differentiate two usernames on case, e.g. BOB and bob are the same user.

- If you do want BOB and bob to be separate users:
  Set unix_case to true, and lowercase_username to false.
- If you are not sure what you want:
  Set lowercase_username true (unix_case false).

Notes on Case Sensitivity for all Platforms:

- Virtual domain prefixes (the second item in a vdomain line) will be in the case given in dmail.conf, unless lowercase_username is set to true, in which case the whole drop file name is made lowercase.
- Setting lowercase_username to true, will make both DSMTP create/read ALL drop files in lowercase.
- Setting lowercase_username to true, does NOT affect how DSMTP does its user lookup of local users.
- Directory hashing (hash_spool 1 and 2) is always done on the lowercase form of the main part of the drop file name (e.g. the drop_prefix is skipped). E.g. hash_spool 2 for the drop file, dom1_BOB would be to, path\b\o\dom1_BOB
- NWAuth, the authentication module that comes with DMail stores names in nwauth.txt file in a case sensitive form, but always does a case insensitive lookup - so we reccommend setting lowercase_username to true so that you do not end up with separate drop files, BOB and bob on Unix based platforms (it does not matter on NT).
- If you are using your own **external authentication module**, you should realise that DSMTP will always do a case sensitive lookup, and it is up to you to decide if your module does a case sensitive lookup. If it doesn't then you should probably set lowercase_username to true to avoid mismatched drop files.

Notes on NT Authentication:

If you have authent_method set to nt_user or you are running the new beta authentication module, ntauth then you are using windows NT's system user database. This is the one that you would normally modify with the User Mananger tool, under 'Programs/Administrative Tools (common)' on your start menu.

Notes:

1. DPOP and DSMTP need to be running with the privilege 'may act as part of the system' in order to be able to check or lookup users. Note dwatch normally runs as a service on NT as the 'system account' user and spawns DSMTP and DPOP, so they to run as that account.
2. When a user logs on to DPOP it has to logon the user to check their username+password combination. To do this the user has to have one of the privileges, 'log on locally' or 'Access this

computer from the network'. This means that all mail users can log on to the mail server machine. If this is a problem you should use another authentication method, e.g. NWAuth

---

DMail Servers' Complete Settings Reference List

This is the complete list of settings used by DSMTP, DPOP and DList which is generated automatically when DMail is compiled.

Many settings in this list will never be fully documented, as they will not be needed by all but one or two customers. We tend to add them at a very fast rate and often for very specific situations :-)

So if a link does not work, then please feel free to contact us at DMail Support. We will be happy to explain the use of the setting, and we will document it further at that stage.

You may also find that you can find more information on a new setting by searching for it on the updates page.

NB: The defaults shown are the code defaults, i.e. for when you have NO setting in dmail.conf, often DMSetup will have inserted recommended 'defaults' for you when it installed DMail. See the individual descriptions for such defaults.

Setting	Default Value	Description	**Version Added**
add_footer	*no default*	Appends the given footers file if the envelope from matches the given domains *(multiple entries permitted)*	2.8a
add_status	true	If true DPOP adds a status line with read or unread.	?
alias	*no default*	wildcard list of aliases for enclosing domain *(multiple entries permitted)*	2.9
alias_fallthrough	true	Tells DSMTP whether or not to use global aliases as a fallback.	2.8q
alias_file	*no default*	Specifies file where DSMTP is to look for user alias information.	?
alias_file_domain	*no default*	Specifies file where DSMTP is to look for domain specific, user alias information. *(multiple entries permitted)*	?
allow_dds	false	Allow ../ and ..\\ in usernames	2.9?
allow_ip	*no default*	*no description available*	?
allowdup	true	Allows mutltiple deliveries to same drop file in same transaction.	?
alt_drop_path	*no default*	Specifies an alternative dropfile path for a particular domain (use only when not using external authentication). *(multiple entries permitted)*	?
apop_enable	false	*no description available*	?

archive_batch	*no default*	*no description available*	?
archive_days	14	Sets a limit to the number of days worth of archive files to keep	2.8i
archive_dir	*no default*	*no description available*	?
archive_in_dir	*no default*	Activates archiving of incoming mail	2.8n
archive_incoming_to	*no default*	Activates selective incoming mail archiving based on the recipient *(multiple entries permitted)*	2.8n
archive_limit	*no default*	Sets a limit (in kbytes) to the space incoming AND outgoing archive files EACH occupy. Overrides archive_days	2.8i
archive_msg_size	8	Puts a cap on the size (in kb) of a message when copied into either archive	2.8n
archive_out_dir	*no default*	Activates archiving of outgoing mail. If the are no archive_incoming directives, all incoming mail will be archived	2.8n
archive_outgoing_to	*no default*	Activates selective outgoing mail archiving based on the recipient *(multiple entries permitted)*	2.8n
auth_allow	*no default*	Sets various permissions for ESMTP/AUTH success *(multiple entries permitted)*	2.7n
auth_hide	*no default*	Switches off the announcing of ESMTP AUTH under various conditions *(multiple entries permitted)*	2.8a
auth_nocache	false	*no description available*	?
authent_cache	1000	Number of authentication requests to cache, External authentication only.	?
authent_domain	false	Set to true to have 'user@domain' instead of 'user' to be passed to the external authentication process	?
authent_method	nt_user	The method used for user/password lookups Unix_User, NT_User or External *(this setting requires a restart for changes to apply)*	? *(Windows only)*
authent_method	unix_user	The method used for user/password lookups Unix_User, NT_User or External *(this setting requires a restart for changes to apply)*	? *(unix only)*
authent_number	2	Number of concurrent authentication processes to run (external authentication only) *(this setting requires a restart for changes to apply)*	?
authent_process	*no default*	Specifies the executable to use for external authentication process(when authent_method external) *(this setting requires a restart for changes to apply)*	?
authent_timeout	10	Timeout (in seconds) for external authentication requests.	?
ban_ip	*no default*	Specifies an IP address that DSMTP should not talk to. *(multiple entries permitted)*	?
ban_mailfrom	*no default*	Specifies patterns for banned mail from lines *(multiple entries permitted)*	2.8b

ban_rcptto	*no default*	Specifies patterns for banned rcpt to lines *(multiple entries permitted)*	2.8b
bin_path	*no default*	Path for user directories containing 'Bin' (i.e. DPOP's ordered drop file format) and msg index files.	?
bin_pfull	0.5	Criteria for compacting. Bins with less than bin_pfull get compacted.	?
bind_in	*no default*	Sets the IP number dsmtp should bind it's listen socket to	2.7j
bind_out	*no default*	*no description available*	?
bomb_dec	50	Mail Bomb; Specifies how much to decrement the entries in the mail-bomb cache by. *(multiple entries permitted)*	?
bomb_dir	./bomb/	Mail Bomb; Specifies which directory mail-bomb messages are to be stored in.	?
bomb_entries	2000	Mail Bomb; Specifies how many entries to store in the mail-bomb detection cache.	?
bomb_max	50	Mail Bomb; Specifies DSMTP's tolerance for detecting a potential mail-bomb.	?
bounce_body	false	Specifies whether or not to include the message body in a Deliver Status Notification message (DSN).	?
bounce_maxlen	20	Specifies (if bounce_body is true) how much of the body (in kbytes) to include in a Deliver Status Notification.	?
brandon	false	Special setting, dsmtp shuts down when cannot write .tmp file.	?
bulletin_from	Email system administrator	Text to be sent as from line in all bulletins. Default is 'Email System Administrator'.	?
bulletin_hasheaders	false	If true then DPOP expects you to give all headers when entering bulletin. Finish headers with a blank line before body!	2.8l
bulletin_max	233	*no description available*	?
bulletin_path	work_path/bulletins	Directory to contain bulletin messages of form nnn.txt.	?
check_gid	*no default*	(UNIX only) Set if group id for drop files should be checked before accessing drop or bin files. Default is no check and set no group access. *(this setting requires a restart for changes to apply)*	?
check_owner_disable	false	(Unix only) Set if user id for drop files should not be checked. Default is to check.	?
check_regulardrop	true	Tells dsmtp to check that dropfiles are regular files	2.8i
cleanout_disable	false	*no description available*	?
crlf_stored	false	Is cr and lf stored at end of lines on this system.	? *(unix only)*
crlf_stored	true	Is cr and lf stored at end of lines on this system.	? *(Windows only)*

delay_badrcpt	false	send a DSN instead of a 5xx response for invalid/non-existent local users	2.8h
dlist_accept_from_request	false	Added for Backwards compatibility. If true then accept messages from usernames containing '-request' (default is false).	?
dlist_backup_size	*no default*	(defaults to 1 Mbyte).	?
dlist_domain	*no default*	Sets the default @xxx address for messages from DList. Overriden by individual lists 'domain' setting (default is first host_domain setting).	?
dlist_log_level	info	Sets the logging level to one of error,info or debug.	?
dlist_names	*no default*	*no description available*	?
dlist_path	*no default*	The directory for lists.dat and all DList work files, log file, and list archiving.	?
dlist_robot	*no default*	*no description available*	?
dlist_rotate	60000	Size of log file in bytes before it is rotated (default and minimum value is 60 kbytes)	?
dlist_smtp_host	*no default*	Use this setting to set the port that dlist should talk on, ipaddress:port (DList will ALWAYS send out messages to the ip address,127.0.0.1).	?
dlist_web_abs	*no default*	*no description available*	?
dlist_web_rel	*no default*	*no description available*	?
dns_disk_disable	false	If true disables the new system for caching DNS lookups to disk	2.8i
dns_host	*no default*	Set specific DNS servers to be used for domain lookups (instead of using system DNS settings). *(multiple entries permitted)*	?
dns_switch_nfails	7	Sets the number of fails before switching dns servers. (-1 is on every fail)	2.8r
dns_timeout	10	Specifies how long (in seconds) DSMTP should wait on a DNS lookup.	?
domain	*no default*	drop file prefix to use for this domain	2.9
domain_banfile	*no default*	*no description available*	?
domain_block	*no default*	*no description available*	?
domain_chroot	*no default*	(UNIX) Robots; Enables chroot functionality for domains. *(multiple entries permitted)*	?
dotstuff_robot	false	Makes DSMTP dotstuff robots for backwards compatibility.	? *(unix only)*
dotstuff_robot	true	Makes DSMTP dotstuff robots for backwards compatibility.	? *(Windows only)*
dpop_host	*no default*	The TCP/IP address of the host DPOP is running on.	?
dpop_maildir	false	If set Maildir format drop files will be scanned by DPOP only, for use in converting from maildir.	2.8r

dpop_path	/usr/local/dmail/	Installation directory for DPOP and manual pages etc.	? *(unix only)*
dpop_path	\\dmail	Installation directory for DPOP and manual pages etc.	? *(Windows only)*
drop_connection	*no default*	Specify an envelope command/string match condition to drop the connection on *(multiple entries permitted)*	2.8h
drop_ext	*no default*	Change-over compatibility; Drop file extension. Default is none.	?
drop_kill	*no default*	Change-over compatibility; File related to drop file to remove after drop file has been burst.	?
drop_max	*no default*	Specifies how big DSMTP should let dropfiles get (in kbytes).	?
drop_old	*no default*	*no description available*	?
drop_path	./mail/	Specifies the directory to use for email drop files	?
drop_prefix	true	If true then the virtual domain prefix is used as part of path for drop files.	?
drop_type	*no default*	*no description available*	?
drop_users	*no default*	Specifies users who need to be able to read mail from another popper as well as DPOP (i.e. convert from bin file back to drop file). *(multiple entries permitted)*	?
dsmtp_path	./	Location of DSMTP executable (and default for many path settings).	?
dtablesize	*no default*	Allows setting of DTABLESIZE above our suggested maximum of 1024	2.8d
dump_stats	60	Time in minutes between performace statistics output	2.7d
dwatch_path	./dwatch/	Specifies path for .pid and .wat files and other DWatch information	?
etrn_relay	*no default*	Gives a list of IP numbers for DSMTP to relay any ETRN requests to *(multiple entries permitted)*	2.8b
expire_age	366	When expiring old emails from server, remove messages more than expire_age days old	?
expire_mins	*no default*	Start a new expire set every this many minutes, unless last one still processing	?
expire_size	1000	When expiring old email from server, remove messages larger than expire_size bytes	?
external_processor	false	Activates DSMTP's message processing daemon functionality	2.8i
external_viruschecker	*no default*	Tells dsmtp to use an external virus checker on MIME attachments from extract_mime	2.8n
extract_mime	*no default*	Tells dsmtp where to extract particular mime attachments *(multiple entries permitted)*	2.8n

fake_verify	true	If true (default) DSMTP returns 250 OK on all VRFY commands.	?
fallback_address	*no default*	Enables a 'catchall' address for a specific domain. *(multiple entries permitted)*	?
files_per_session	2	Number of file handles needed per dpop session. Setting <2 may allow more concurrent sessions. Set between 1 and 2 use less than 2 at your own risk!	?
filter_skip	*no default*	Gives a wildcard username that triggers message filter bypassing *(multiple entries permitted)*	2.8i
filter_skip_onlyif	*no default*	Gives a wildcard username that triggers message filter bypassing only if all names match either skip or skip_onlyif *(multiple entries permitted)*	2.8i
forward	*no default*	Creates a mail re-direction rule to be checked against all incoming mail. *(multiple entries permitted)*	?
forward_cc	*no default*	Creates a carbon-copy mail re-direction rule to be checked against all incoming mail. *(multiple entries permitted)*	?
forward_from	*no default*	Anti-SPAM; specifies a domain that is exempt from relaying restrictions. *(multiple entries permitted)*	2.4h
forward_from_ip	*no default*	Anti-SPAM; specifies an IP address that is exempt from relaying restrictions. *(multiple entries permitted)*	?
forward_user	*no default*	Anti-SPAM; turns on an anti-relaying system that allows relaying only after a recent DPOP login.	?
forward_window	120	Anti-SPAM; sets time (in seconds) in which relaying is allowed after POP login (i.e. expiry time of forward_user records).	?
from_test	false	If true use slow test for valid From lines when bursting files. Only needed when DPOP is used with other smtp servers which use extended From line test.	?
fromip_drop	false	Tells dsmtp to drop connections from IP numbers who have exceeded fromip_max	2.8n
fromip_max	10000	Anti-SPAM; sets a limit on message throughput per hour from any IP number.	?
fromip_nolimit	*no default*	Anti-SPAM; specifies IP addresses which are exempt from the fromip_max setting. *(multiple entries permitted)*	?
gateway	*no default*	Bypass DNS lookups by specifying domain-IP_address pairs for DSMTP to use. *(multiple entries permitted)*	?
hash_qfiles	false	Activates directory hashing of qfiles	2.8i
hash_spool	0	Sets hashing method (0,1 or 2). Hashing is where drop_files are distributed across multiple directories. *(this setting requires a restart for changes to apply)*	?
hide_rcvd_ip	*no default*	Makes DSMTP hide the specified IP address in the 'received' message header.	?

host_domain	127.0.0.1	Adds a domain name to the list of domains to be recognized as being local *(multiple entries permitted)*	?
imapd_include_level	*no default*	Sets the time in minutes between retries.	2.8q(4.3.3q)
imapd_mailbox_name	*no default*	If 'name' specified, IMAPD puts .mbx files in ~user/name (not required for most setups).	?
kpop_port	*no default*	*no description available*	?
language_file_dsmtp	*no default*	Specify a language file for dsmtp to use	2.8h
lock_id		(UNIX); sets a file locking id for multi-server systems.	?
log_chain	false	*no description available*	?
log_data	false	Specifies whether or not DSMTP should log the TCPIP data it gets.	?
log_days	14	Tells DSMTP how long (in days) to keep summary logfiles.	?
log_flush	false	*no description available*	?
log_level	error	Specifies how much information to output to the logfile (one of error, info, debug).	?
log_milliseconds	false	Tells DSMTP to log millsecond-accurate time	2.8f
log_mime	false	Switch MIME info logging on	2.8h
log_path	./log/	Specifies where the server log files are to go.	?
log_status	600	Time (in mins) between logging DPOP status to log file.	?
lookup_names	false	Sets DSMTP and DPOP to do reverse DNS lookups.	?
lowercase_password	*no default*	If true DPOP will always set passwords to lower case before using them for authentication (auth module must have lower case passwords or be case insensitive).	?
lowercase_username	false	Sets DSMTP and DPOP to be case insensitive for usernames and hence 'dropfile' names.	?
maildir_althome	*no default*	*no description available*	?
maildir_home	*no default*	*no description available*	?
manager_ip_address	*.*.*.*	IP addresses manager commands can come from (affects tellpop and DMAdmin etc.). *(multiple entries permitted)*	?
manager_ip_name	*no default*	Domain names that manager commands can come from (affects tellpop and DMAdmin etc.). *(multiple entries permitted)*	?
manager_password	*no default*	Password for valid manager commands (OBSOLETE after DPOP version 2.0 - no longer in dmail.conf). *(obsolete)*	?
maps_action	*no default*	Invokes certain actions should the connecting server be registered with MAPS *(multiple entries permitted)*	2.8e
max_log_size	3000000	Max size (in bytes) for DPOP log file before renaming and starting new one.	?

max_loglen	1024	Specifies the maximum size (in kbytes) of DSMTP's log files.	?
max_msgsize	2048	Specifies the maximum message size DSMTP may accept(in kbytes).	?
max_others	20	Specifies the maximum number of recipients recorded for each message in summary files.	?
max_queue	1000	Sets the number of rcpt lines from queue files that DMSTP queues up in memory, this is not a setting to idly play with:)	?
max_rcpts	*no default*	Use to limit the number of recipients DSMTP should allow per message.	?
max_rcpts_session	*no default*	Sets a per-session maximum RCPT count	2.8i
max_rcvd	30	Anti-SPAM; Specifies the maximum number of received lines allowed in an incoming message.	?
max_retry	48	This command is obsolete, use max_retrytime *(obsolete)*	?
max_retrytime	24	Specifies the maximum time in hours to continue attempting (once every 2 hours) to deliver a message.	?
max_send	10	Sets maximum outgoing channels, which limits maximum simultaneous outgoing messages (50 is upper limit).	?
max_sessions	200	Limit on number of concurrent sessions connected at one time, DPOP will try to make this many available.	?
max_vdomain	2048	Maximum number of virtual domains which can be used	2.8b
min_space	10	Specifies the minimum disk space (in Mbytes) DSMTP needs to operate.	?
msg_filter	*no default*	Tells DSMTP to use a message filter (in specified file) on all incoming mail.	?
msg_separator	*no default*	Change-over compatibility; Message separator character if not defined then we don't use one.	?
msgid_suffix	*no default*	Specifies a msgid suffix (instead of host_domain)	2.8c
no_autohost	false	De-activate the auto adding of hosts to DSMTP's internal host_domain list.	?
no_dotforward	false	(UNIX) tells DSMTP not to look for .forward files but .fwd files instead.	?
no_xdpop	false	*no description available*	?
nt_domain	*no default*	*no description available*	?
oneline_stats	false	Sets the format for stats file loglines.	2.8p
orbs_action	*no default*	Invokes certain actions should the connecting server be registered with ORBS *(multiple entries permitted)*	2.8b
pop_event_log	burst,stat	Switches on logging of various POP events: last,uidl,burst,list,listn,stat,user,pass,quit,retr,dele,rset *(multiple entries permitted)*	2.8b

pop_port	110	Allows the the port number pop3 clients will access to be set to a non-standard value.	?
pop_timeout	600	How long DPOP waits (in seconds) before assuming a connection has gone and close TCPIP channel.	?
prefix	*no default*	drop file prefix to use for this domain	2.9
preserve_domain	false	Added for Backwards compatibility. Do not replace all host domains with the first one for lookups	2.7i
proxy_domain	*no default*	Tells dsmtp to use a proxy when trying to talk to the given domain *(multiple entries permitted)*	2.8n
qfile_split	1000	Tells DSMTP how many address lines to put into a q_idx file before making another one	2.8e
quiet	true	Tells DSMTP to direct minimal output to stdout.	?
quota_fix	*no default*	Zeros user quota when they login, to fix previous fault (turn on for about 1 week only!).	2.7j
ras_domain	*no default*	RAS Dialup; This setting is NOT necessary in almost all cases. It specifies the domain on which authentication is to occur (NT only).	?
ras_entry	*no default*	*no description available*	?
ras_number	*no default*	*no description available*	?
ras_password	*no default*	*no description available*	?
ras_smtpip	*no default*	*no description available*	?
ras_timeout	*no default*	*no description available*	?
ras_timer	*no default*	*no description available*	?
ras_username	*no default*	*no description available*	?
rbl_exception	*no default*	specify ip number exempt from RBL-based actions *(multiple entries permitted)*	2.8h
reject_bad_header	false	Tells dsmtp to reject messages with invalid RFC822 headers	2.8n
reject_no_reverse	false	Tells DSMTP if it should reject messages where reverse DNS on connecting IP address fails (requires reverse_lookup to be set).	?
relay_to	*no default*	Permits unconditional relaying to particular domains. *(multiple entries permitted)*	?
remind_timeout	3600	Specifies the minimum time (in seconds) between critical error emails.	?
retr_chunk	10000	Max bytes to send in one chunk for retrv command. Default is 10000.	?
retry_interval	120	Sets the time in minutes between retries.	2.8p
retry_invalid_domain	false	Tells DSMTP to continue trying to deliver to a bad domain name, rather than bouncing	2.8r
reverse_name_ban	*no default*	Specifies domain names that should be banned (only works if lookup_names is true) *(multiple entries permitted)*	2.8b

robot_defaultuser	*no default*	Sets an alternative uid for robots attempting to run as root	2.8i *(Windows only)*
robot_defaultuser	0	Sets an alternative uid for robots attempting to run as root	2.8i *(unix only)*
robot_try	120	Specifies how long DSMTP should try to give input to a robot before giving up (in seconds).	?
robot_wait	600	Specifies how long DSMTP should wait before killing a robot.	?
rotated_logs	4	Sets the number of old log files dsmtp keeps.	?
scripted	false	Experimental command. Currently less than alpha. DON'T DOCUMENT.	2.8b
self_rcvd	3	set a limit to the number of times dsmtp will send a message to itself	2.8h
shell_prefix	*no default*	Robots; tells DSMTP to use a prefix for autoresponder exec calls.	?
show_8bitmime	false	Makes DSMTP revert to pre version 2.5c default of advertising 8 bit MIME capability (not recommended).	?
show_ehlo	*no default*	Switches on the announcing of various ESMTP extenstions *(multiple entries permitted)*	2.7j
signal_core	*no default*	If true, DSMTP will not intercept signals such as segmentation fault(UNIX based platforms only).	2.8d
slave_burst_size	1000000	Burst drop files of this size (in bytes) or larger with a DSlave process.	?
slave_number	4	Sets the number of DSlave processes (sub processes of DPOP) for handling burst of large drop files (we recommend 4).	?
slave_process	dpop_path/dslave.exe	*no description available*	? *(Windows only)*
slave_process	dpop_path/dslave	*no description available*	? *(unix only)*
slave_timeout	100	Timeout (in seconds) for commands (e.g. burst) given to DSlave processes. Default 100 seconds.	?
smart_reload	false	Use smart (buffering) reloading code	2.7e
smtp_welcome	*no default*	A template for the welcome line. Accepts \\r\\n delimited lines, and $DATE, $HOST macros	2.8a
spam_dump	*no default*	Tells DSMTP to dump filtered spam to the given filename, rather than the bit-bucket *(multiple entries permitted)*	2.8e
spool_dir	*no default*	Activates a mail spooling directory for incoming mail *(multiple entries permitted)*	2.8n
stats_path	*no default*	Path for log files of per connection usage statistics,if set turns on statistics logging.	?
stop_listen	false	on shutdown stop listening for new connections	2.8f

suffix	*no default*	username containing this suffix match enclosing domain	2.9
suspend_domain	*no default*	Permanently suspends a domain's q files (unless requested by an ETRN) *(multiple entries permitted)*	2.8b
sysadmin	*no default*	The email address of the system administrator and 'postmaster' for DSMTP.	?
tarpit_except	*no default*	Lists IP addresses which are exceptions to tarpit_start. *(multiple entries permitted)*	?
tarpit_start	*no default*	Anti-Spam; Number of Recipients before DSMTP should start responding slowly.	?
tcp_list_nodelay	false	*no description available*	?
tcp_max	200	Limits the total number of TCPIP connections. Includes both incoming and outgoing connections.	?
tcp_nodelay	false	*no description available*	?
tcp_send_size	65000	tcp output buffer size	2.8d
tcp_timeout	300	Specifies how long DSMTP should wait (in seconds) for a response before giving up on ALL TCPIP connections (except sendlog connections).	?
tellpop_host	*no default*	The address for tellpop to use to talk to dpop.	2.8p
timezone	*no default*	Tells DSMTP to fake its timezone info with this value.	?
to_ip	*no default*	dpop connections from this ip match enclosing domain	2.9
unix_case	false	(UNIX)Tells DSMTP to use strictly case sensitive user lookups.	? *(unix only)*
use_drop	false	When use_maildir true Tells DPOP to check for drop files as well	2.9?
use_drop_dir	*no default*	*no description available*	?
use_flock	true	If true use flock. Use false on nfs systems if flock always fails. Use false with lockid so dotlock is used instead	2.8d
use_forward_files	true	Used to stop DSMTP from checking for users' forward files.	?
use_maildir	false	*no description available*	?
user_ip_address	*.*.*.*	IP addresses user connections can come from. *(multiple entries permitted)*	?
user_ip_name	*no default*	Valid domain names user connections can come from. *(multiple entries permitted)*	?
user_quota	false	Enables a per-user mailbox quota system and optionally sets the default quota(true/false/kbytes).	?
users_file	work_path/users.idx	*no description available*	?
valid_users	*	Valid usernames to get their mail here. *(multiple entries permitted)*	?

vdomain	*no default*	Sets up a virtual domain for use with/by DPOP and DSMTP. *(multiple entries permitted) (this setting requires a restart for changes to apply)*	?
vdomain_passwd	*no default*	Tells DSMTP and DPOP to use a separate passwd file for a vdomain (Unix) *(multiple entries permitted)*	?
vdomain_separator	_	Specifies the separator character to use with Virtual Domain prefixes.	?
vdomain_substitute	true	Allows DSMTP to do 'smart' string replacement of host_domains with appropriate vdomain in message headers.	?
virtual_user_post	*no default*	Adds a sendmail style virtual user table, actioned immediately before lookup *(multiple entries permitted)*	2.8b
virtual_user_pre	*no default*	Adds a sendmail style virtual user table, actioned before any other local-user rules apply *(multiple entries permitted)*	2.8b
warn_user	4	Specifies when (after x tries) DSMTP is to warn the sender of a delayed delivery.	?
work_path	./work/	Specifies the directory for work files (also default for log and statistics files)	?
xdpop_header_hide	false	*no description available*	?

Additional Help and Support

Every purchase of a DMail License Key includes twelve months free email support. We pride ourselves on the quality of the support we provide for our software and answer most questions within twenty-four hours (please always resend your message if you do not get a response within that time).

The contact email address is:
support-dmail@netwinsite.com
(be careful not to get that confused with the DMailWeb support address support-dmailweb@netwinsite.com)

We also have a user mailing list to enable users of our software to share experiences and insights and to provide us with feedback on what new features are of general interest.

To join the DPOP/DMail mailing list send an email message to dmail-list-request@netwinsite.com with a message containing the line
subscribe
You can then send messages to the list by sending to dmail-list@netwinsite.com

All mail ariving from the DMail list is marked in the subject line with [dmail-list], e.g.
Subject: [dmail-list] does anyone know . . .

Purchase also includes free upgrades to any new release of DMail during the twelve months following date of purchase.

Contacting Netwin

Products Downloads Prices Support Company

The Tellsmtp Command Line Utility

The Tellsmtp executable provides a simple universal command line utility for controlling and monitoring the DSMTP system. Of special note is the 'reload' command, which makes DSMTP reload its configuration file. It is often used after manually editing the config file, so that you do not have to restart DSMTP for the changes made to take effect.

It is invoked with the command line:

tellsmtp [-n ip] [-p port] [-i conf] <command> [parameters...]

The **ip** parameter following the **-n** switch tells Tellsmtp which IP number to use when connecting to DMail. The default is simply to connect to the machine Tellsmtp is executed from.

The **port** parameter following the **-p** switch tells Tellsmtp which port to use when connecting to DMail. The default is 25.

The **-i** switch tells Tellsmtp to look for appropriate settings in the DMail config file specified by the **conf** parameter (including full pathname).

The <**command**> parameter can take the following forms, as outlined below.

Tellsmtp Command Summary:

```
clear_cache_all - tells DSMTP to clear its cache of user lookups (and the forward
rules etc. that they returned).
config <setting> - shows what DSMTP currently has set for that 'setting'
filters - shows all active mail_filter, or filter_file commands
help             Displays a brief summary of Tellsmtp commands
log_level       sets the amount of information logged
monitors        shows the IP numbers dsmtp is currently remote logging to
profile        - shows a profile of code statistics.
quiet           toggles 'quiet' status, i.e. does/does not print minimal operating
information to stdout
reload          tells DSMTP to reload its config file
register        instigates registration function, creates register.txt
resume          tells dsmtp to resume processing outgoing mail
scriptfile.msc  sends the script file named
sendlog         tells dsmtp to send live log info
showchans        shows a list of current TCPIP channels and their status
showsend        shows a list of current outgoing TCPIP channels.
shutdown        starts exit of DSMTP in a reasonably civilized fashion
smtp            what follows this command is passed directly to DMail
status          causes status of DSMTP to be echoed to screen
suspend         stops dsmtp from attempting to process outgoing mail
tryall          requeues all outgoing messages to be processed immediately
```

Detailed List of Tellsmtp Commands:

---

clear_cache_all

tells DSMTP to clear its cache of user lookups (and the
forward rules etc. that they returned).
See authent_cache.

---

help

Displays a brief summary of Tellsmtp commands.

---

register

When given the register command, Tellsmtp will execute
the DMail registration function. This asks questions which will allow DMail
to be registered. A register.txt file is created for emailing to Netwin.
Payment details are encrypted for safe transfer to Netwin Ltd. No parameters
are required.

---

scriptfile.msc

If the **<command>** parameter ends in .msc, Tellsmtp
assumes that a Tellsmtp scriptfile has been specified. It will then load
the scriptfile, and pass the lines in it directly to DMail. Each line will
be sent to DMail exactly as it is found in the textfile. The only exception
is when the line starts with '#'. In this case, Tellsmtp will *not*
wait for a response before sending the next line. This is mostly for use
when sending the DATA lines of a message. Because of this flexibility,
a scriptfile may be written to test any aspect of DMail's operation. A
simple scriptfile to deliver a message locally would read as follows:

```
helo test.domain
```

```
  mail from:<test@any.machine>

  rcpt to:<recipient@local.machine>

  data

  #Subject: This is a
test

  #

  #Please ignore

  .

  quit
```

The hash symbol must be present at the beginning of all lines between a DATA command and the '.' following it, or Tellsmtp will wait (forever) for a reply, and DMail will wait (forever) for more data. Assuming the above script is saved as 'test.msc', the command

```
  tellsmtp test.msc
```

will send the scriptfile to DMail (or whoever is listening to the port Tellsmtp is talking to). No parameters are required.

---

smtp

The smtp command talks directly to DMail, with Tellsmtp pretending to be an SMTP client. Any arguments following the smtp command will be concatenated (separated by a space) and sent to DMail, followed by a quit command. DMail's responses will be echoed by Tellsmtp. This command is of little use, as it will only work with a very small number of SMTP commands.

Example:

```
  tellsmtp smtp helo test.domain
```

---

status

This command will cause Tellsmtp to echo status information sent to it by DMail. Depending on what DMail has been up to, several screens of information may be displayed. It is a good idea to redirect Tellsmtp's output to a file when using this command, e.g.

```
tellsmtp status > a.a
```

outputs the status to the file, a.a.

If DMail is behaving oddly, but not crashing, the output from a Tellsmtp status command may help us to find the problem.

Here are some details of the information that it prints out. Don't worry if you can't understand some of it.

```
+DATA
 DSMTP v2.7l status as at Tue Nov 09 12:35:05 1999
 Uptime: 0 days, 16 hours, 48 minutes

 Message Statistics
Messages     received: (number of messages received by dsmtp during the uptime
                                                    period)
Messages local/remote: (number of msgs with both local + remote rcpt lines)
Messages    delivered: (number of queued files delivered locally)
Messages      sent on: (number of queued files delivered with rcpt lines that
                                                    were all for remote addresses)
Messages      gave up: (number trashed for some reason (unkown rcpt etc))
Messages      pending: (number still sitting in the message queue)
Messages   to unknown: (other people talking to this server gave this many bad
recipient lines)
Messages   incomplete: (messages where dsmtp could not get the entire message)
Messages     rejected: (number of messages rejected for whatever reason)

Total in:  (the total number of messages that arrived at dsmtp's door)

Total out: (a ballpark figure of the total number of messages that dsmtp dealt with)


 File Rty Time Left Busy Domain
(This is a list of queued files which are currently in
the memory queue as set by the setting max_queue)
-----------------------------------------------

15236  000 -942172505    1 aol.com
15215  000 -942172505    1 aol.com
15188  001 000007140    0 companyx.com

(using the last one as an example,

15188: is the queue files number, i.e. you will find
q_15188.itm and .idx files in the work_path

001:  rcpt lines in this queue file have been tried once
```

000007140: time in seconds until this is next tried. If a
large negative number this message will be tried as soon as
dsmtp can get to it - e.g. if you do a tellsmtp tryall command
all messages will be set like this.

0: Is the queue file currently being looked at, 0 is NO, 1 is Yes.

companyx.com: the domain of the first rcpt line in the queue file, to
give you an indication of who it is addressed to.
)

 TCP channel history

(The column on the left is a counter of the number of times
each close/open of a TCPIP channel occurred for the detailed reason,
some reason relate directly to a setting in dmail.conf as noted below.)
19717 Command failed: RCPT Unknown user

77598 Connection opened:(in)

68534 Connection closed:(in) it was told to (other end gave QUIT command)

53729 Connection refused: banned IP number (connecting ip address matched a ban_ip setting)

12144 Connection closed:(out) dsmtp told it to (dsmtp does this when finished with a normal connection)

2936 Connection closed:(in) by remote end (other end closed the connection on dsmtp)

7692 Connection closed:(in) timeout (dsmtp waited tcp_timeout seconds for a response and then closed connection)

3461 Connection closed:(out) unexpected isclosing() (connection was aborted, other end crashed, line broke ...)

15604 Connection opened:(out) processing qfile (dsmtp opens connection as q file was to non-local domains)

1436 Command rejected: remote domain, no relaying (got a non-local rcpt line from a remote domain and dsmtp coudl not allow it to relay)

014 Command rejected: wierd RCPT (bad syntax in rcpt to line)

013 Command rejected: untimely RCPT (order fo SMTP protocol was not followed by sending device)

8054 Command rejected: too many RCPTs (number of rcpt lines greater than setting, max_rcpt)

287 Connection refused: accept() failed (remote end opened channel, dsmtp responded but could not connect to other end)

294 Command rejected: wierd FROM (bad syntax of from line given)

060 Command rejected: DATA no valid rcpt fields (rejected connection at data stage because no valid destination addresses were given)

1150 Command rejected: DATA fromip_max limit hit (sending ip address had sent more than allowed messages this hour, as set by fromip_max)

006 Command rejected: premature FROM (sender got SMTP protocol around the wrong way)

003 Command rejected: DATA no valid from field (we did not get a valid FROM field but the sender still tried to send)

001 Command rejected: Found possible MX loop (message had a lot of rcpt lines to a domain that resolved to this box but we did not have a host_domain or vdomain setting for it)

004 Command rejected: unsupported AUTH (sender tried to use an AUTH type that dsmtp does not support)

.

+OK finished

---

showchans

This diagnostic command will make DSMTP respond with a list of TCPIP channels that have been used since startup. List entries show the channel's current state and how long the channel has been in that sate. Channels which will not timeout (e.g. those
sending live log updates, see sendlog) are marked as such.

The basic layout is:

```
chan[v] used=w, state=x, socket=y, ip=a.b.c.d,
                last active z seconds ago
```

The used and state columns are not of use to sysadmins so simply ignore them.

Here are three lines from an example output with explanation:

```
chan[0] used=2, state=1, socket=132, ip=127.0.0.1,
                last active 8 seconds ago (receiving loglines)
```

```
chan[1] used=2, state=6, socket=133, ip=199.99.99.4,
                last active 1 seconds ago

...

chan[150] used=0, state=0, socket closed, ip=199.99.99.3,
                dead for 8652 seconds
```

The first line shows that something is connected from the IP address, 127.0.0.1, on DSMTP's first channel using the operating system's TCPIP socket 132. At the end of the line the words '(receiving loglines)' indicate that it is a connection from something that has asked dsmtp to copy any lines that it logs out to it on this . In this example it is the DMAdmin windows GUI admin tool that is connected. The line also shows that something (DSMTP or DMAdmin) talked on that port 8 seconds ago.

The second line is an active TCPIP connection from an email client or SMTP server at IP address 199.99.99.4.

The last line shows that a maximum of 150 channels have been opened at some point. Because it is marked as 'socket closed' you know that that connection is no longer open, and dstmp has not had to use that channel again since something connected from the IP address 199.99.99.3 8652 seconds ago.

Inactive connections will timeout after the time specified by the setting, tcp timeout.

---

shutdown

This command will cause DSMTP to exit in a reasonably civilized fashion, but will cause it to do so without any further confirmation from anyone (i.e. immediately)

---

die

This command will cause DSMTP to crash. Used mainly for testing how well it recovers from...ermm...a crash. Use at your own risk. Actually, try not to use at all :-)

---

reload

This command will make DSMTP reload its config file. This
is very useful for making changes to DSMTP on the fly, without having to
shut it down and restart it. Alterations to significant commands such as
drop_path can be made, so take care when using this command. The only setting
which will have no effect if reloaded is smtp_port.

NOTE : If you have modified a dmail.conf setting which
is relevant to both DPOP and DSMTP, it is necessary to reload **both**
servers individually with their reload configuration file commands, e.g.
tellsmtp reload **and** tellpop reload. DMAdmin
will do this automatically, but note that you can send the reload command from the
command list
in DMAdmin whenever you wish.

If you have an exceptionally large dmail.conf file this process of reloading could
take 30
seconds. So larger sites should probably schedule reloads for every x hours and then
let
customers know that their changes won't take effect until after the next scheduled
reload.

---

log_level

This command will change DSMTP's log level during runtime.
It will not cause DSMTP's config file to be re-written, so next time DSMTP
is run, it will use the log level setting found there. This command is
useful for closely observing a particular transaction DSMTP may be about
to make.

The available log levels are:

error:	the only information written will be errors, warnings, socket read and write information and minimal progress information.
info:	as well as the error information, this setting gives much more progress information, as well as file open and close calls.
debug:	as well as the info information, this setting gives a whole lot of internal status information, function calls...all sorts of stuff. However, it may slow down operation slightly and can produce large log files, so use sparingly.

If DSMTP is behaving unusually, run it with the verbose option first. This should give enough information to reveal whether or not it's a problem with the config file. If all else fails, run it with the debug option. If DSMTP is crashing or doing very peculiar things, the most useful things to Netwin support are the config file and the log file.

Info is the default setting, and the usual setting for operating in. For more information see the **Logging of information and error messages** section.

---

quiet

This command toggles DSMTP's "quiet" status, i.e. whether or not it prints anything aside from minimal operating information to stdout. On some systems (Windows NT in particular) this will result in significant performance gains. If DSMTP is running quietly, critical messages such as disk full warnings will still be displayed.

---

showq <num>

This command tells dsmtp to send the q_<num>.idx and q_<num>.itm files back to tellsmtp.

---

killq <num>

This command tells dsmtp to kill the q_<num>.* files
and remove them from its internal queue. Use this in conjunction with **showq**
to remove troublesome or unwanted messages.

---

suspend

This command will stop dsmtp from processing outgoing
mail. dsmtp will stay in this mode until told otherwise, so use with caution,
as incoming mail will still be accepted, so a backlog of q files will
build up.

---

resume

This command will tell dsmtp to resume outgoing message
processing. It undoes what **suspend** does.

---

sendlog

This command will cause DSMTP to send a copy of all its
logfile output across the TCP connection opened by the sender of the sendlog
command.

---

tryall

This command tells dsmtp to requeue and immediately retry
all pending outgoing messages. Useful mainly if there has been a network
outage of some form.

---

monitors

This command will list all current sendlog TCPIP connections, e.g. connections from remote admin tools such as DMAdmin.

Back to Top of
Command Summary List

Products Downloads Prices Support Company

DMail Robots

DSMTP can send a mail message to a program or "robot" for it to do with it as it pleases. Once the message has been given to the robot DSMTP considers it delivered, and consequently if the robot cannot be run or does not take the message then DSMTP will bounce the message.

Note: If the robot looses the message it is its fault :-) and DSMTP will not notice.

An example of a robot is an autoresponder, which automatically sends a message back to the sender of the message for the recipient. We have provided a version of an autoresponder, called DRespond in the DMail distribution set.

To run a robot you need to create a Special Forward. Typically you create a forward setting, that specifies that a message received by DSMTP should be delivered to a specified robot. To let DSMTP know that the message is for a robot you specify the the robots path starting with the pipe symbol, |, in place of the destination address.

forward john@big.com |drespond.exe

You can turn any mail redirection feature into a 'Special Forward'. So as well as a forward setting you could have an alias file entry (alias_file or alias_file_domain) or use a '.forward' file or use the 'fwd="" field in the response to a user database lookup.

Examples: To deliver all mail for john@big.com to an autoresponder called, drespond.exe you might set up an alias like,
john: |drespond.exe
or using the fwd field a user lookup might return,
+OK john@big.com config 0 fwd="|drespond.exe"

Note: To include command line arguments for the robot, put the entire command line within quotes, e.g. forward john@big.com "|drespond.exe arg1 arg2" and all multiple word arguments should also be within quotes, e.g.
forward john@big.com "|drespond.exe "more than one word" arg2"

Aliases and forward rules do not deliver the message to the original recipient, so if you want the original recipient to get the message as well as the robot, then you will have to use a forward_cc setting, e.g. forward_cc john@big.com |drespond.exe or for an alias or fwd field use the keyword, '$USER' as one of the destinations, e.g.
john: $USER,|drespond.exe
+OK john@big.com config 0 fwd="$USER,|drespond.exe"

Autoresponders - DRespond

An autoresponder accepts a mail message from the SMTP server, analyses who it is for and who it was

from and then sends a preset message back to the sender. Often the preset message says something like - john is on holiday, he will respond to your message when he gets back.

DRespond

DRespond is Netwin's autoresponder robot. We provide it in a ready built form on all platforms. If it is not in your distribution set then please contact DMail support.

Basically to activate DRespond for a user, you should create a forward rule or an alias that pipes the incoming message to drespond, e.g.
In dmail.conf add the following forward rule:

> forward bob@domain1.com "|\dmail\drespond.exe \dmail\message.txt -from "postmaster@domain1.com" -subject "AutoResponse:Bob is on holiday." "
>
> will forward any mail for the user bob@domain1.com to drespond which will feed DSMTP an autoresponse message to the sender of the message.

OR, add an alias line like this to an alias file:

> bob@domain1.com: "|\dmail\drespond.exe \dmail\message.txt -from "postmaster@domain1.com" -subject "AutoResponse:Bob is on holiday." "
>
> will forward any mail for the user bob@domain1.com to drespond which will feed DSMTP an autoresponse message to the sender of the message.

OR, in version 2.5d and above, you can enter the robot in the fwd="" field returned by an external authentication module:

> +OK username config 0 fwd="destination_address,|c:\dmail\drespond.exe c:\dmail\message.txt -hasheader"
> See Ext. Auth. FWD Field for details.

Notes:

- In all cases you need to use the pipe symbol, |, in the destination address.
- You should include the pipe and all command line arguments within quotes (except for in the Ext. Auth. FWD Field).
- Multiple word command line arguments should also be within their own quotes (this is not possible when using the Ext. Auth. FWD Field option - you must use the -hasheader argument).
- If you want the original recipient to receive the message, e.g. if they are on holiday and wish to read all of their mail when they return, then you should use a forward_cc rule instead, i.e. forward carbon copy (when using the Ext. Auth. FWD Field option you can do this by using the $user symbol as a destination address).
- If you want someone else to deal with their mail while they are away then you should put in a separate forward rule to the other person, e.g.
  forward bob@domain1.com "|drespond.exe . . .
  forward bob@domain1.com fred@domain3.com
  so that fred at domain3.com gets all of bob's mail while he is away, but the autoresponder will

answer it too.

Further Details:

DRespond takes a text file as its main argument. The text within this file is used to form the body of the message to be sent as the autoresponse. If you use the -hasheader option then you can specify the From: and Subject: headers in the message text file, they should be the first two lines and then there should be a blank line between them and the message body.

DRespond can be run from the command line, which is a good way to check that it is doing what you require, e.g.

Usage: drespond file.name [-debug] [-hasheader] [-subject "string"] [-from person@domain]

where

file.name	is the name of the file to be used as the response message body (including path relative to the drespond executable if necessary). NB you should include the FULL path.
-debug	causes drespond to create a log file, of executable_name.log, e.g. drespond.log in the same directory that the executable is running from.
-hasheader	tells drespond that the message file (file.name) already has the From: and Subject: headers in it. If this is not specified then these two headers should be supplied as command line arguments, see below.
-subject "string"	specifies the Subject: header of the response as 'string' if -hasheader is not specified. NB remember to use quotes for mutliple word strings.
-from "string"	specifies the From: header of the response as 'string' if -hasheader is not specified.

NOTE: currently drespond cannot take a mapped drive path in the message file name on windows NT, e.g. f:\shared\message.txt or \\shared\message.txt do not work. We think that this is a windows NT bug because of the robot being spawned as a sub-process by DSMTP - it works when the robot is run from the command line. If you are more enlightened than us :-) or have got it to work, then please contact support-dmail@netwinsite.com and let us know.

Here is an example message text file, file.name, with the from and subject headers included:

```
From: postmaster@domainx.com
Subject: The person you emailed is currently on holiday . . .

This is the message body, note the blank line above separating it from the message headers
and it goes on to say that the person is away until the end of the week but will still get their
mail (because you have used a forwarding option where the original recipient gets the
message as well as the DRespond robot) and that they will respond to it when they get back.
```

Forward Rules, Aliases and Gateways

Overview

DSMTP has four ways of setting up an alias or mail redirection.

We DO NOT recommend using a mixture of ALL of them :-)

You can:

1. create a forward rule (in dmail.conf)
2. create an alias rule (in a domain specific aliases file)
3. create a 'forward' file for each user (in the drop directory or home directory)
4. return a new destination address in your external authentication response

Skip straight to some examples
How to forward multiple copies

In general the syntax of all of the above rules require an 'alias_address' and a 'destination_address'.
E.g. a forward rule might look like,
forward bob@domain1.com julie@domain2.com
which has the syntax:
forward alias_address destination_address

When DSMTP gets an incoming message it looks for a rule with an alias_address that matches the address on that incoming message (the 'RCPT TO:' line of the SMTP protocol). The rule then gives DSMTP a new 'destination_address' to which DSMTP tries to deliver the message.

So what are the basic differences?

1. Forward settings are simple because they are processed as soon as a message arrives - i.e. before anything else is done. So they can be for ANY domain, not just domains on your server. They must contain the full email address (both user and domain) for both the alias_address and destination address. E.g.
forward Bob.Smith@domain bsmith@domain

Forward settings are entered in dmail.conf itself (or you can use #include if you want them in another file).

2. Similarly an alias might be:
Bob.Smith: bsmith@domain
i.e. aliases have the syntax,
alias: destination_address
Aliases are only for users on local domains (as set by host_domain and vdomain settings).

You enter aliases in an 'alias file' that is specific to the domain that the user is in.

3. You can also have a forward file for any specific user. This forward file is checked just before the incoming message is written to the user's drop file. The file can be located in the user's home directory (on UNIX based platforms) or in the drop file directory.

Since the file is for a specific user, it only needs to contain a list of (comma separated) destination addresses.

Because the forward file is looked for so late in the process DSMTP will only find and action it if no other redirection rules have already been found (unless they allow the original recipient to also get the message).

4. The last option is available to you if you are running an External Authentication Module. You can make your module (NWAuth has it built in) return a new destination address in the fwd="" field of its response.

This is very nice as you can edit forwarding in your user database (e.g. LDAP or ODBC databases).

Below are some general notes on mail redirection. These are followed by the following sections:

Notes:

1. DSMTP does not necessarily look for the different types of rule at the same point in the message delivery process.
2. In general you can use a mixture of the different mail redirection options, but we do not recommend using all of them at once as the expected result can be difficult to work out. See mixed forwarding systems
3. You can also in general have multiple rules for any given alias_address and they will all be applied, e.g.
   forward mymail destination1
   forward mymail destination2
   etc.
4. All methods of mail redirection can be used to make DSMTP deliver messages to a specific file or to a robot, see Special Forwards.
5. Forwarding implies that the original recipient does not get the message. However most of the forwarding options have a method to allow the original recipient to get the message as well as the new destination. E.g. instead of using a forward setting you could use a forward_cc setting.
6. For information on checking forward rules see the following FAQ,
   How can I check what aliases I have set up for a user?

---

1. Forward Settings

Forward settings are for any domain, local or non-local users. DSMTP searches for and applies them as soon as it receives a message.

E.g.
forward bob@domainx.com julie@domainy.com will cause DSMTP to forward any mail received for the address, bob@domainx.com to the destination address, julie@domainy.com.

If DSMTP cannot get the message delivered to the destination address, i.e. julie@domainy.com, then it will bounce the message in the normal way.

The received address, bob@domainx.com above, can be for any domain. It does not have to be one that DSMTP recognises as a host_domain or vdomain setting.

NB: Notes on forward settings:

- The original recipient does not get the message. If you want them to, use the forward_cc setting.
- Forward settings can be created for any users on any domain
- There is a fallback_address setting to pick up mail not covered by a forward setting for a specific domain.
- If you have domain names with upper case characters in them, then you should create forward settings for them in the lower case. This is because DSMTP will lower case the domain name before looking for settings in dmail.conf.
- After changing a forward setting you should only have to do the tellsmtp reload command, to make DSMTP reload its forward settings.

Relevant settings: (in the reference section)
forward forward_cc

---

2. Aliases

An alias is written on one line, starting with the user part of an address that DSMTP should look for on incoming messages (in the RCPT TO: line). Then you type a colon followed by a space and the destination address that DSMTP should deliver the message to instead, if it finds the match. Here is an example,
Bob.Smith: bsmith@domain
i.e. alias: destination_address
(the two addresses are separated by a colon, ':' and a space)

You add the entry to a file that contains aliases for a specific domain, i.e. you create a domain(s) specific alias file.

To do this add a setting to dmail.conf like,
alias_file_domain domain filename
where domain is the domain to which all of the aliases in the file, filename, apply.

After changing aliases remember to reload DSMTP with the tellsmtp reload command.

NB: Notes on Aliases:

- NOTE: All aliases found for a user by DSMTP are applied.
- Aliases are only for local domains. DSMTP checks that the destination domain is local before looking for an alias.
- Aliases are case sensitive except for the Postmaster alias, which in version 2.4e and above, can be any case (POSTMASTER, postmaster,PoStMaStEr etc.).

  So for the examples above you must have separate aliases for
  Bob.Smith: bsmith@domain
  and
  bob.smith: bsmith@domain

  There is a fallback_address setting to pick up mail not covered by a forward rule for a specific domain.
- The postmaster alias should not be specified in an alias file as DSMTP automatically creates it for all domains. The destination is set by the dmail.conf setting, sysadmin
- After changing or adding an alias within a file you should only have to do the tellsmtp reload command, to make DSMTP reload its aliases.
- In versions 2.5e and above you can specify the alias as one of the destination addresses, if you want the original recipient to also get the message, e.g.
  bob.smith: bob.smith,user@domainx.com
  will now work. This is known as a 'self referencing alias'.

Relevant settings: (in the reference section)
alias_file alias_file_domain

---

4. Ext. Auth FWD field

External Authentication modules can optionally return the field fwd="emailaddress" in their response to a successful 'lookup' or 'check' command, e.g.
+OK username config 0 fwd="destination_address"

If such a field is returned, then DSMTP will cache the destination address along with the username. DSMTP will then use the destination address as a forwarding rule for that user. So any mail arriving for that user will be forwarded to the destination address, as if there was a forward rule for them in dmail.conf. Obviously then this type of forwarding is only for local domains, where dsmtp is going to try to look up the user.

In Version 2.5d and above you can make your external authentication return multiple destination addresses, as well as a destination file or robots and also the $user symbol, so that the original recipient receives the message (like forward_cc). E.g. the following examples are now valid;

single destination address:
+OK username config 0 fwd="destination_address"

multiple destination address:
+OK username config 0 fwd="destination_address1,destination_address2"

single destination and original recipient receives the message:
+OK username config 0 fwd="$user,destination_address"

forward to destination and to a robot:
+OK username config 0 fwd="destination_address,|c:\dmail\drespond.exe c:\dmail\message.txt -hasheaders"

forward to specific file:
+OK username config 0 fwd="c:\dmail\filename"

The multiple entries are comma separated (spaces are not a separator), and you CAN NOT have quotes within the forward field. This means that you cannot put multiple word arguments in the command line for the robot (as they would need quotes around them). See the drespond -hasheaders option to help get around this problem.

The $user can be used once with any other destination in the field.

The tellsmtp command,
tellsmtp clear_cache_all
was added in version 2.4e, to reset DSMTP's cache.
(we will be adding tellpop clear_cache_all and tellpop clear_cache user at a later date)

DSMTP caches the last 1000 lookups. So after changing a FWD entry in the external authentication database you have to do a,
tellsmtp clear_cache_all
command, from a command prompt.
Both DPOP and DSMTP cache their user lookup results. On a password fail, they refresh their cached entry, just in case the password has changed for the user. So you should not need to do a cache clear for either DPOP or DSMTP after changing a password.

You should look at the source to NWAuth to get an idea of what should be returned by your external authentication module. Also see the External Authentication section for details on the syntax of the external authentication protocol. Note that the fwd="" field is an optional field which comes after the uid.

FYI . . .
If you are wanting to do multiple forwards or to pipe the message to a robot, and you do not have a version where you can enter multiple destination addresses in the fwd="" field. Note that you can enter a destination_address that points to multiple forward rules in dmail.conf. Similarly the forward rule(s) could pipe to a robot.

---

3. Forward Files

The other method for forwarding messages is using forward files for specific users. Forward files are checked just before the message is written to the user's drop file. If found then the message is not written to the user's drop file and the redirection is done instead.

For the rest of this section note that
user_drop_name
is the users drop file name, which may, depending on your settings, include a drop_prefix, e.g. dom1_username and may have a vdomain_separator other than the underscore '_'.

Note: DSMTP has been made to work with all types of forward file so if you mix forwarding systems then we appreciate that DSMTP's operation gets rather complicated. Basically we recommend deciding on one way to do forwarding and stick to it. For example it is not recommended to let some users do forwarding with a forward file and let other users use the fwd field of external authentication. At the end of this section there are some notes covering what happens when you use mixed systems.

On Windows platforms:

To set up a forward file for a user you should create a file, user_drop_name.fwd in the same directory as the user's drop file.

In this file you should add a list of comma, space or semicolon delimited addresses. If it is a pipe to a robot or drop file, then the entries must be one per line.

For example a .fwd file could look like this:

bob@domainx.com,julie@domainy.com
dog@domainz.com;cow@domainq.com
horse@domainw.com rabbit@domainn.com
"|\dmail\drespond.exe message.txt"
"\strange_path\username"

Which would result in all of the above addresses, robots and strange drop files receiving the message (but as with all forward rules the original recipient won't get the message - see forward_cc if you want the original recipient to receive the message). Normally you would probably only have one or two entries in a forward file.

Notes on forward file syntax:

- entries can be separated by being entered on a new line
- entries are separated on a line by commas or semicolons (not spaces)
- special forwards should be entered on a line by themselves
- special forwards can have quotes around them if required

On UNIX based platforms:

You can use either:

1. UNIX style .forward files where the file is located in the user's home directory

2. Our .fwd files which are located beside the user's drop file (as described for windows platforms above).

You can only choose option 1 if you are using UNIX authentication, i.e. you have set
authent_method unix_user
and you have set home directories in the system password file.

If not using External Authentication and the passwd file check finds a valid user, then DSMTP will exclusively look for the file ~user_drop_name/.forward, and use it if it finds it. Where ~user_drop_name is the home directory for the user from the passwd file. If it cannot find the .forward file then it won't look for any other forward files, but if the home directory entry is blank then it will look for the .fwd file as per the windows platform details above. Note that virtual domain password files set with the vdomain_passwd setting are processed in the same way, i.e. you can enter a home directory for the users and DSMTP will look for a .forward file within it.

If you have home directories specified for users in the passwd file (or vdomain passwd files) but want the user's forward file in the drop file directory, then you can set
no_dotforward true
and DSMTP will only ever look for a .fwd file as per the windows platform details above.

If you are using external authentication then DSMTP will look for the .fwd file on UNIX based platforms (as per the windows platform details above).

Notes on using Mixed Forwarding Systems:

DSMTP has been made to work with all types of forward file so if you mix forwarding systems then we appreciate that DSMTP's operation gets rather complicated. Basically we recommend deciding on one way to do forwarding and stick to it. For example it is not recommended to let some users do forwarding with a forward file and let other users use the fwd field of external authentication.

Forward rules, aliases and the fwd="" field returned as part of an External Authentication are separate forwarding systems that have **precedence** over forward files, i.e. If any of these exist for a user DSMTP will not look for a forward file. Having said that all the other forwarding options have precedence, a forward file will still affect them if they use a self referencing option, i.e. where that system calls for the original recipient to also receive the message - e.g. $user in a fwd="" field. In these cases DSMTP has been told to try to deliver the message to the user (amongst other things) and just before it writes the drop file it checks for the forward file and if found, uses it INSTEAD of delivering to the original recipient.

Here is all the above in a semi-coded form for those of you who prefer to read it this way . . .

When dsmtp uses .forward . . .

DSMTP checks or does the following and they MUST ALL be true:
authent_method Unix_user (so not ext. auth)
no_dotforward false

password lookup succeeds
password lookup returns a home directory

If they are all true then dsmtp checks the file,
home_directory/.forward

When dsmtp uses .fwd . . .

if any of the above are NOT true, AND the external authentication returns an EMPTY fwd="" field OR one with $user, then it checks the file,
drop_file.fwd
where drop file includes the drop_path (possibly including hashing) and the drop file name (possibly including drop_prefix and vdomain_separator), e.g.
d:\dmail\in\t\y\dom1_bob.fwd

To clarify, if the external authentication does return a forward destination then the .fwd file is not used UNLESS the destination is self referential - e.g. includes, $user. If the destination does include $user but the .fwd file exists then the original recipient still won't get the message.

Special Forwards

All of the forwarding methods allow you to make DSMTP forward the message to a special destination of a specific file or program. To do this...

for a robot:
you must use the pipe symbol, '|' at the start of the destination address.

for a file:
you must include a path character, '/' or '\', which means that you must specify a full path and filename.

E.g. for a forward rule to write the file to the special UNIX file, /dev/nul, (it makes things disappear!) you could have,
forward bob@domainx.com /dev/nul

In this manual we refer to programs that receive a message (read in on stdin) as a 'robot'.

So similarly to send Bob's mail to a robot, called drespond, you might have,
forward bob@domainx.com |c:\dmail\drespond

If you want to make DSMTP take command line arguments for the program or robot then you should include the destination address within quotes, e.g.
forward bob@domainx.com "|c:\dmail\drespond arg1 arg2"

For more information on robots see
Robots and specifically on autoresponders, see
Autoresponders - DRespond

Routing

DSMTP can be used to route mail on to another server. To do this you will probably need to use gateway and forward settings and possibly, relay_to.

To help you work out which setting to use this is the very very basic outline of what DSMTP does with a message when it arrives:

1. apply any forward rules
2. check if the destination is a local domain (if not then queue it up for relaying on to the correct destination server (a gateway setting can be used for non-local domains that will often be encountered, so that DSMTP does not have to do a DNS lookup)
3. (is a local domain) looks for an alias
4. then do a lookup on the username (not the password)

The alias, or the forward or the username lookup (using an external auth module) can all specify that the message should go to a different destination, or to a robot. But given that they don't then DSMTP writes the message to the local user's drop file.

Gateways

DSMTP has a gateway setting, which has the syntax,
gateway domain ipaddress

If DSMTP is going to do a DNS MX lookup on the domain specified, then it will instead simply use the ipaddress specified.

In general this means that the gateway setting will be used for outgoing mail or for incoming mail that is not destined for the local server (relaying).

FYI . . .
The SMTP protocol was originally intended to allow messages to take an indirect path from SMTP server to SMTP server, in order to reach the server where the user is considered 'local', this is message relaying. These days a message will normally go directly from the sending user's local SMTP server directly to the SMTP server where the destination user is 'local'. This is because the DNS mail exchange (MX) lookup done on the destination domain, will normally return the IP address of the end server and because server administrators tend to not allow mail to pass through their email server (there is generally no need and it allows spammers to hide).

If you want messages to go via a specific IP address, then you can use a gateway setting to accomplish this. This is often done to allow mail to be sent through fire walls or where you have distributed internal email servers and you want email to enter the internet through one point.

Bounces and Delivery Status Notification (DSN)

(under construction, 16/6/99)

What happens when DSMTP cannot deliver a message?

The answer to this depends on what stage the message has got to before DSMTP cannot deliver it.

Note: You may like to read the FAQ, Tell me about the SMTP protocol

1. The connecting client is thrown off the server because of where they are connecting from - e.g. a ban_ip setting.
2. At the Rcpt To: stage of the SMTP protocol.

   DSMTP responds with a 500 level SMTP code and a text message giving the reason for rejection.

   This is the best place for DSMTP to reject the message as the sender will normally get an immediate message from the email client showing DSMTP's reason for rejection, e.g. the user is not known as a local user or relaying is not permitted by them.
3. At the end of the DATA stage:

   DSMTP responds with a 500 level SMTP code and a text message giving the reason for rejection.
4. when a relayed message is rejected by the next server:

   DSMTP creates a DSN if it has been requested, otherwise it bounces the message.
5. when a message for a non-local server cannot be delivered YET:

   DSMTP queues the message and sends a DELAY DSN if requested. It then tries to send it again in 2 hours time. It continues to try to send it up to the number of hours set by the max_retrytime setting.

   Examples of reasons for a message not being able to be delivered are, the DNS server is not responding (so the destination server's ip address cannot be resolved), the destination server is not responding or returns a 400 level SMTP code (meaning come back later).
6. when DSMTP has accepted a message for local delivery and then fails to write it to the local user's drop file:

   This is very unlikely.

   DSMTP would create a DSN message if requested or otherwise bounce the message.

Settings Affecting Bounces:

Bounces are affected by the following DSMTP settings in dmail.conf:
bounce_body
bounce_maxlen

DSNs:

Delivery Status Notifications can be requested as part of the Extended SMTP (ESMTP) protocol.

This is done on the end of the RCPT TO: line of the SMTP protocol with the NOTIFY command. E.g.
RCPT TO:<bob@domain> NOTIFY=FAILURE
requests that the sender wants to be notified of a failure to deliver to bob@domain.

The NOTIFY command can take the value NEVER or one or more of, SUCCESS,FAILURE,DELAY.

Some examples:
RCPT TO:<bob@domain> NOTIFY=SUCCESS
RCPT TO:<bob@domain> NOTIFY=FAILURE,DELAY

Where:

+ A NOTIFY parameter value of "NEVER" requests that a DSN not be returned to the sender under any conditions.

+ A NOTIFY parameter value containing the 'SUCCESS' or 'FAILURE' keywords requests that a DSN be issued on successful delivery or delivery failure, respectively.

+ A NOTIFY parameter value containing the keyword 'DELAY' indicates the sender's willingness to receive 'delayed' DSNs. Delayed DSNs may be issued if delivery of a message has been delayed for an unusual amount of time, but the final delivery status (whether successful or failure) cannot be determined. The absence of the DELAY keyword in a NOTIFY parameter requests that a 'delayed' DSN NOT be issued under any conditions.

See also, I got a bounce (Delivery Status Notification) message from DSMTP ... for a list of explanations of common DSN and bounce messages.

Mail Re-direction Examples

1. **How to redirect but do the normal delivery as well:**
   (e.g. fred wants a copy of bob's mail, but bob still wants to receive it too)

   Mail re-direction usually means that the original recipient does not get the message, but it is easy to instruct DSMTP to carry out the delivery as well as the re-direction.

   a) Using Forward settings: use a forward carbon copy.

   forward_cc bob@domain1.com fred@domain2.com

   results in mail addressed to bob@domain1.com being re-directed to fred@domain2.com AND still going into bob's mailbox, bob@domain1.com.

   b)In alias files: use the special destination, $user, in the alias file destination.

   bob: $user,fred@domain2.com

   results in bob getting his mail as normal and fred@domain2.com also getting a copy.
2. **How to forward multiple copies:** (fred and john want to receive all mail addressed to sales)

   a) use multiple forward settings as all matching forward settings will be applied.

forward sales@domain1.com fred@domain1.com
forward sales@domain1.com john@domain2.com

results in fred and john receiving mail addressed to sales@domain1.com and the sales mailbox does not get any mail (fred and john could be at any destination domain).

b)In alias files: give a comma separated list of destinations.

sales: fred@domain2.com,john@domain2.com

results in fred and john receiving mail addressed to 'sales' instead of it going to the 'sales' mailbox. By specifying the domain to which the alias file applies you can control what domains this alias exists for. See the next example.

3. **How do you set up an alias on all local domains?**
(bob wants to get mail addressed to sales@*)

   If bob really wants mail addressed to the user 'sales' at any domain which passes through DSMTP (incoming AND outgoing) then simply use a forward setting,

   forward sales@* bob@domain1.com

   Note - if bob tries to send a message to sales@othercompany.com then he will find his outgoing message gets re-directed back to him!

   More likely you only want bob to get mail addressed to sales at LOCAL domains.

   Option 1:

   If you add a domain as an alias of the main domain, by adding host_domain settings after your main domain, e.g.

   host_domain domain1.com
   host_domain domain2.com
   host_domain domain3.com

   then the user sales, will automatically have the aliases,

   sales@domain1.com
   sales@domain2.com
   sales@domain3.com

   so then all you have to add is an alias to redirect mail from sales@domain1.com to bob@domain1.com. One way to do this is to add the following alias to the domain1.com alias file,

   sales: bob@domain1.com

   (use the following line to tell dsmtp about the domain1.com alias file,
   alias_file_domain domain1.com c:\dmail\dom1aliases.txt)

   Option 2:

You can use alias files to create the sales alias for any local domain, by specifying an alias file with the wildcard character.

The alias_file_domain setting has the syntax,

alias_file_domain <domain> <filename>

You can specify that the file contains aliases for all domains with the wildcard character, i.e.

alias_file_domain * c:\dmail\global_aliases.txt

If you just want the alias for domains, domain1.com, domain2.com and domain3.com then point 3 settings at the same file, i.e.

alias_file_domain domain1.com c:\dmail\aliases_group1.txt
alias_file_domain domain2.com c:\dmail\aliases_group1.txt
alias_file_domain domain3.com c:\dmail\aliases_group1.txt

The domains, domain1.com, domain2.com and domain3.com can be declared as local with either a host_domain or vdomain line.

In both cases use the following line to add the alias in the alias file you have specified

sales: bob@domain1.com

ETRN

Yes, DMail supports the ETRN protocal as part of its support for Extended SMTP, ESMTP. This allows you to host or gateway for domains which are located on a 'remote' or 'dial up' server.

Basically, if another SMTP server sends DMail the ETRN command,e.g.,
ETRN domainx.com
DSMTP will try to send any mail in its outgoing queue waiting for that domain.

NB: Dmail handles the queueing of mail for a 'dial up' domain automatically as part of its normal message queueing. If a connection cannot be established to the destination server (i.e. the dial up server that is one of your ETRN domains) then DSMTP queues the message and tries to send it every 2 hours up to a limit set by the setting,
max_retrytime.

So in theory to host an ETRN domain you don't have to do anything, DMail does it automatically.

However in practice you probably need to set a,
gateway
setting for the domain and you will probably also need to set your SPAM rules so that users can relay mail through your server to that domain. The easiest way to do this is generally with the,
relay_to
dmail.conf setting.

We have recently added a setting to version 2.8b,
suspend_domain
which can be useful. You can set it to suspend all mail for a specified domain, so that dsmtp does not try to send the mail every 2 hours. This setting does not affect the ETRN command, so that when an ETRN is received for the domain, DSMTP will still send all mail queued for it. NB: This can be a little dangerous because if a domain did not dial up and send ETRN for 2 months, any waiting mail would not get bounced to the sender.

DSMTP can also send the ETRN command, which allows it to act as the 'remote' or 'dial up' server. It sends the ETRN command as part of the ras_timer RAS dial up setting.

DMail Frequently Asked Questions:

No. 1 question:
How do I set up a 'HotMail' type system?

Questions:

Answers:

1. Drop users:

   You have a few users who check their mail using a normal POP client but leave the mail on the server and want to be able to access the drop files directly, with pine for example. But DPOP converts the drop files to its own format for more efficient manipulation, so once the mail has been checked there is nothing left in the drop files and the users can't see their mail. This is easily remedied by adding a line to your dmail.conf configuration file. It should look like this:

   drop_users ralph,bill,*smith

   This would force DPOP to leave all the email messages for ralph, bill and anyone with a usercode finishing with the word smith, in drop files. Be careful not to put spaces in the list and avoid making it too general as there is a performance hit in keeping messages in drop files, that's why DPOP avoids it in the first place. This setting is only needed for users who check their mail with a POP3 connection AND leave it on the server AND want to read it with software that directly reads the drop file.

   ---

2. What operating systems is the DMail package available on?

   It is our intention to make it available on all common operating systems. Initially available on Linux, Solaris, HPUX and Windows NT. Please ask if you need it for another system soon.

   ---

3. What is the maximum number of email clients which can be handled by DPOP?

   This basically depends on the server hardware it is to run on and the type of license you buy.

It is intended to be very scaleable and to work well on large and small systems. Because of its design both large numbers of concurrent users and large numbers of email user accounts have relatively little impact on the process size and performance.

---

4. We have our own special username/password routines. Can these be used with DPOP/DSMTP?

   Yes, DSMTP and DPOP can be configured to use an external authentication process for checking username/passwords.

---

5. Is the source available so that we can tailor it to our needs?

   No, but this should not be necessary as most aspects of DSMTP DList and DPOP can be easily configured. They can also use an external password checking routine, an external routine to indicate where drop files are and how the path is hashed. DPOP can also generate statistics which can be used by an external routine for generating charging information. If there is some other aspect which you need to be able to tailor please let us know.

---

6. We would like to try DPOP but are paranoid about upsetting umpty thousand users. How can we ease into it?

   Email is a vital service so even if the current popper you are using is slow it is still a scary step to move to another one. You can't afford to upset users. So how do you ease into it. There are a number of strategies which can be helpful here.

   - If you have the luxury of a spare machine obviously installing DPOP on that first will help. It at least allows you to check out the various options you might want to use and get used to how they work. The DMSetup wizard will help you to remove it from the test machine after your testing is complete. The de install option tries to err on the conservative side. It tells you where the files are that you might want to delete. It will only remove something that is definitely part of DPOP and not any other popper.
   - If you have not got a spare machine or you have tried that and are now more comfortable but still cautious: The next easy step is to install DPOP on the main server BUT get it running on a different port. This way you can leave your original popper running. For example you might set DPOP up on port 1100 instead of 110. To do this, follow the normal installation procedure but say no to the question: "Shall I comment out current POP3 entries in inetd.conf". Then edit dmail.conf file and change pop_port line as shown below:
     pop_port 110
     pop_port 1100
     You can then get individual users to try switching to DPOP use by changing the setting in their email reading software to read on another port. This is straightforward in Pegasus mail, more difficult on some other email clients. For Eudora on Windows 95 just edit the Services file in the windows directory to change POP3 port. You can even allow someone to connect both ways although if they are going to do this AND leave unread or undeleted mail on the server you must put a line in dmail.conf to tell DPOP to change their bin files back into a drop file at the end of each session. This

should only be done if they NEED to read their mail from Unix command line or some other non DPOP connection. It will slow processing down. If Bob,Bill and Bert are Unix gurus who read their mail from the Unix command line and using a POP3 client, you might add one of the following lines to dmail.conf:
drop_users B*
drop_users Bob,Bill,Bert
Once you have run DPOP in this mode for a while you can switch back to the real POP3 port by changing the pop_port line in dmail.conf and then issuing the Tellpop reload command.

- Alternatively you can take the plunge and install DPOP directly on your main server in some off peak time. Test it with a few test accounts and if there are any problems that look difficult, revert to the previous popper. To do that all you need to do is put the lines in inetd.conf back how they were and get inet to reload. The DMSetup wizard can do this for you. If the accounts you have tested have undeleted or unread mail left on the server these must be converted back to drop files. This must be done before stopping DPOP by using either:
tellpop drop_all
to do all accounts that have used DPOP or
tellpop drop Bert
tellpop drop Bill
etc. to deal with user accounts one at a time.

---

7. Should I use username suffixes or multiple IP numbers for virtual domain support?

   Multiple IP numbers has the advantage that the users do not need to change their username setting in their email client packages. Username suffixes save you having to configure your server machine to respond to multiple ip numbers. The two schemes work as follows:

   If a vdomain setting line has an IP number like 1.2.3.4 in it then DPOP checks what ip number the user was connecting to and does stuff based on matching vdomain lines. If the vdomain setting line has a suffix string rather than an IP number in the same place (e.g. /xusers) then when users connect to DPOP and sends user fred/xusers DPOP picks up the /xusers and uses that to match a vdomain line. The suffix is stripped off and the prefix is added just as it would be for an ip based vdomain. From then on the two systems are the same. The other question is what do we end up with as a drop file name.

   Consider the two vdomain lines:

   - vdomain abc 1.2.3.4 xdomain.com /var/spool/mail/xdomain
   - vdomain abc /xdom xdomain.com /var/spool/mail/xdomain

   If a user connects to 1.2.3.4 or uses a username fred/xdom
   Then the Unix username used will be

   - abc_fred

and the drop file used will be

- /var/spool/mail/xdomain/fred

Some mail transport systems find it easier to deliver to a drop file

- /var/spool/mail/xdomain/abc_fred

To allow for this another setting has been added

- drop_prefixes true/false

if this setting is true DPOP will use the second form for the drop file name.

---

8. Can I setup a 'HotMail' like system using DMail or DMailWeb? (Technical details on WAdduser)

   Yes, we have a Web Based Email system that offers Auto Account Creation. For general information on such systems see, Setting Up Web Based Email System with Auto Account Creation
   Our OLD way of doing this is presented below...

   Yes, using wadduser instead of NetAuth you need:

   - cwmail (web to mail interface)
   - dmail (dsmtp,dpop)
   - nwauth (external authentication module for dmail)
   - wadduser (example web cgi for adding users using nwauth)

   Note: You no longer have to use WAddUser with our new product NetAuth.

   DMAIL comes with source and binary examples of nwauth and wadduser, you should examine the source and modify wadduser.htm so that it only allows the users to automatically create their own accounts (it has extra functions which you would not want them to be able to do)

   Technical details:

   1. Fetch the source for nwauth/wadduser. This should come with dmail but if you have an earlier version you can download it from
      ftp: //ftp.netwinsite.com/pub/netwinsite/dmail/nwauth.zip
   2. Make any changes to the source that you want (not required)
      See How do I add extra fields to wadduser? for some more information on this.
   3. Building wadduser.cgi and nwauth (only needed on UNIX)

      Unix:
      gcc wadduser.c nwauth.c -DNOAUTHMAIN -o wadduser.cgi
      rm nwauth.o (so you can build it without NOAUTHMAIN defined)
      gcc nwauth.c -o nwauth
      Note: if you get crypt errors you may need to add, -lc -lcrypt to the end of

each gcc line.

Windows:
Create two console (command line) projects,
1 builds nwauth.exe from nwauth.c,
2 builds wadduser.cgi from both wadduser.c and nwauth.c but you need to define NOAUTHMAIN as a preprocessor definition.
NB:In both projects you will probably need to add wsock32.lib to the list of standard linked libraries.

4. Install the cgi script and the html form

   windows:
   copy wadduser.cgi \inetpub\scripts (or wherever your web server cgi directory is)
   copy wadduser.htm \inetpub\wwwroot

   Unix platforms:
   cp wadduser.cgi /home/httpd/cgi-bin (or wherever your web server cgi bin directory is)
   cp wadduser.htm /home/httpd/htdocs
5. Test the cgi, use netscape and reference your web site:
   http://your.web.server/wadduser.htm
   Fill out the form and press one of the buttons, if it fails, you will probably need to modify the 'action' in wadduser.htm
6. Tell dmail to use nwauth for user authentication, add or change in dmail.conf (/etc/dmail.conf or \winnt\system32\dmail.conf)

   authent_method external
   (unix) authent_process /usr/local/dmail/nwauth
   (NT) authent_process c: /dmail/nwauth.exe
   authent_number 1
7. Modify wadduser.htm so it only allows the actions that you want users to be able to perform, (e.g. not delete or search)
8. On UNIX you will need to set some file protections:

   touch ..../cgi-bin/adduser.log
   chown nobody .../cgi-bin/adduser.log
   touch /usr/local/dmail/nwauth.txt
   chown nobody /usr/local/dmail/nwauth.txt
9. If you wish add a bulletin message to DPOP that welcomes all new users.
10. You can add a file, added.htm, in your cgi directory and wadduser will display the contents of the file when a user has been successfully added - underneath the 'Adding User' title.

---

9. **I want all domain1 email which does not go to a specific user to go to one designated user.**

   The setting you want is fallback_address, e.g.
   fallback_address domain1 default@domain2

   FYI . . .
   I gather that you were using forwarding rules to try to do the same thing instead of using the fallback address. I note that from the lines you had set up, you seemed to be expecting DSMTP to stop looking through the list of forward rules when it found the first match. So for example you had something like,

   forward bob@domain1 bob@domain2
   forward fred@domain1 bob@domain3
   forward *@domain1 default@domain4

   and expected DSMTP to only action the bob@domain1 line if a message came in for bob@domain1, i.e. you wanted the *@domain1 line to 'catch' any messages that did not match the first two forward rules.

   The way DSMTP has been written, all forwarding rules that are found to match for an incoming message are applied and forward rules are also applied instead of delivering the mail to the original recipient. So if a message came in for bob@domain1 given the dmail.conf lines above, bob@domain2 would receive the message AND so would default@domain4 (because both of the forward rules can be matched) BUT bob@domain1 would not receive the message.

   Whereas the fallback address setting,
   fallback_address domain1 default@domain4
   does what you want. I.e. if a message came in for bob@domain1.com and it could not be delivered, because the user database did not have an entry for bob and there wasn't a setting (forward rule, alias etc.) sending the mail to someone else, then DSMTP would deliver it to the fallback address, default@domain4, instead of bouncing the message back to the sender.

   Note: DSMTP's action of applying all forward rules is a nice feature that you will probably use for other situations.

---

10. **What is Relaying?**

    Sending mail to non-local users is referred to as 'relaying', as DSMTP must relay the message to the user's local SMTP server (often their ISP's SMTP server) so that it can write the message to the user's drop file (mail file on the server).

    The message may be relayed several times from server to server until it reaches the final SMTP server where the user is a local user - at least that is the theory. Because of spammers, most SMTP servers severely restrict what relaying is allowed to occur. So the message normally only gets relayed through an intermediary SMTP server if the server the email

client gives the message to for sending is setup to gateway mail to another server, i.e. pass all its mail onto that server for delivery. An SMTP server set to gateway mail is often used to allow mail to be sent through fire walls.

---

11. **How do I add extra fields to wadduser?**

To add extra fields in wadduser.htm for storing more information about the user, you will need to do the following:

- Add the input text boxes and their appropriate variables in HTML to wadduser.htm (or the pages that you want them on)
- Modify the source of the CGI wadduser (wadduser.c) so that it records the information given
- Recompile wadduser.c (which requires linking to nwauth.c)
- Replace wadduser.exe in your cgi or scripts directory with your new version

The page that calls the wadduser CGI (wadduser.htm) has a form on it that calls the CGI as its action to perform when it is submitted, i.e when one of the buttons is pressed. E.g. action="http://server.com/scripts/wadduser.exe" calls the wadduser cgi from the scripts directory on the server.com web server. The CGI works out which of the buttons on the page was pressed and carries out the appropriate action.

The function below web_add (from wadduser.c) is called when you click on the "add" button on the example wadduser.htm page.

The form also has a number of variables that are passed to the CGI as part of the action of submitting the form, e.g. name, username, password. To add more fields you need to add more such input fields to the web page, in this form,
<input type="text" name=" username" size="20">

So to add a field to get the person's hobby, you could add to wadduser.htm
<input type="text" name="hobby" size="20">

Then you need to decide what you want the CGI to do with the information in the fields that you add.

The three lines in the function below,

fprintf(f,"%s|",form_find("phone"));
fprintf(f,"%s|",form_find("fax"));
fprintf(f,"%s|",form_find("comments"));

search the form that is submitted by the wadduser.htm page for the fields, phone, fax and comments and if it finds them then it prints them into the log file, adduser.log. If it cannot find them, for example if there is no such input field on the web page (this is the case with the example wadduser.htm - there are no input boxes for phone, fax and comments) or the user has not entered anything in the box, then it will simply enter an empty string.

So to make wadduser log the person's hobby entry, you could add this line below the three

above,
fprintf(f,"%s|",form_find("hobby"));

The function below ONLY writes the username, password and name entries to the nwauth.txt password file, but it writes to the log file, adduser.log, a whole bunch of input fields that don't exist. Note that nwauth only takes three fields, 'username', 'password' and 'other'. It is the 'other' field into which you can add your own fields. The function below adds the field 'name' into the 'other' field in the following format,
name="the person's full name"
The 'other' field can take as many fields as you want (until the information reaches the BFSZ definition, when you will get buffer over flows!) simply make sure that each field has the correct format and that they are separated by a space.

So to make the CGI write the hobby field onto the end of the 'other' field in nwauth.txt you should change the line in the function below from,
```
  sprintf(bf,"name=\"%s\"",name);
```
to
```
  sprintf(bf,"name=\"%s\" hobby=\"% s\"",name,form_find(hobby));
```

This will result in nwauth.txt lines like,
bob:a234h6:name="Bob Smith" hobby="ping pong"
for the username bob, which has a password of something we cannot read as it is encrypted, and a full name of 'Bob Smith' and a hobby of 'ping pong'.

```
int web_add(void)
{
FILE *f;
char username[BFSZ],password[BFSZ],name[BFSZ];
char bf[BFSZ];
/* Check the user has filled in the required fields */
if (!check_value("Name","name","")) return 0;
if (!check_value("Username","username","")) return 0;
if (!check_value("Password","password","")) return 0;

f = fopen("adduser.log","a");
if (f==NULL) { printf("Could not write file\n"); return 0;}
fprintf(f,"%s|Add|",get_date());
fprintf(f,"%s|",mygetenv("REMOTE_ADDR"));
fprintf(f,"%s|",form_find("username"));
fprintf(f,"%s|",form_find("name"));
/* These are optional form elements to record */
fprintf(f,"%s|",form_find("phone"));
fprintf(f,"%s|",form_find("fax"));
fprintf(f,"%s|",form_find("comments"));
fprintf(f,"\n");
fclose(f);
```

```
ncpy(username,form_find("username"),BFSZ-1);
ncpy(password,form_find("password"),BFSZ-1);
ncpy(name,form_find("name"),BFSZ-1);

strlwr(username); /* Only allow lower case usernames */
do_header("Adding user");
printf("<pre>");
if (auth_exists(username)) {
  printf("Sorry, a user by that name already exists\n");
} else {
  sprintf(bf,"name=\"%s\"",name);
  auth_set(username,password,bf);
  showfile("added.htm");
}
printf("</pre>");
do_footer();
return 0;
}
```

---

12. Time Stamp and Time Zone problems (mostly on Linux platforms).

    NB: the Date field is normally added to an email by the email client. DSMTP only adds one if the email client has not put one on (e.g. if the message was created by DMail's sendmail stub).

    NB: In version 2.7l DSMTP was changed to add time stamps that are in local time on both the Date header if it adds one and on the Received lines. Before this it always stamped GMT on any Received headers that it added.

    If you are running a newer version of Linux (e.g. RedHat 5.2 etc.) then you may experience problems with the time stamp and timezone in the DMail servers. This is because of the difference in C libraries used to compile DMail. Examples of the problems are, the timezone being incorrectly specified or all time stamps being in GMT.

    To fix the timestamp problems, you need to use a version of DMail compiled with the newer libc6 libraries or have the below fix applied. There are other benefits to the new libraries, e.g. support for shadow passwords etc. and we have been building versions of DMail which use them since version 2.4j. So if you are running a platform that can support the newer libraries then we recommend that you download one, marked 'linux_libc6' from the main or beta download directory,
    ftp://ftp.netwinsite.com/pub/dmail

    The alternative is this fix:
    Create the proper link by executing this command.

    ln -s /usr/share/zoneinfo /usr/lib/zoneinfo

    (Sorry, I'm not sure what version of Unix this answer works on :-(

Also:
On many platforms the timezone information is incorrect so in dmail.conf you can define:
timezone xxxx
This controls the time zone string that DSMTP stamps on outgoing messages, to give it the form,
hh:mm:ss xxxx
NB: it does not alter the time printed, only the timezone string following it.

Some Examples:
timezone +1100 would give 11:30:33 +1100
timezone -0800 PST would give 11:30:33 - 0800 PST
timezone -0600 CST would give 11:30:33 - 0600 CST
timezone +0100 CET would give 11:30:33 +0100 CET
timezone +1200 would give 11:30:33 +1200

---

13. How can I transfer mail accounts (users) from my current email server?

    The best way to answer this is to give you some details on options for DMail and hopefully if you are able to tell DMail support about your current system then they can make relevant suggestions.

    It is worth noting first off that if the users are simply members of the operating system user database then you do not need to do anything with them - simply install DMail and it will find the users by default.

    DMail has two basic authentication options,

    a) use the operating system password list
    b) use an external authentication module

    There is one configuration file, dmail.conf, setting that sets this,
    authent_method

    For a this will either be,
    authent_method nt_user
    or
    authent_method unix_user
    depending on whether you are on a windows or Unix based platform.

    For b you set,
    authent_method external
    and
    authent_process path_to_program
    where path_to_program is the authentication program to run.

    Your options are:

    1. We provide an example authentication module, called NWAuth, which is fully functional and is very efficient with large numbers of users.

2. You can also write your own to link to any type of user database (or modify one of ours).
3. Our example module for linking into an LDAP server, LDAPAuth.
4. Our example module for linking into DNews's users.dat file, DNAuth.
5. A customer has provided us with the source to talk to a mySQL server, which DMail support can pass on to you to use or modify.
6. There is a link on the following page to an ODBC authentication module provided by another customer,
   http://netwinsite.com/dmail/utils.htm

So one of the above might be an option, but it does depend on how the user's details are stored. Our NWAuth module can also be run from the command line, e.g.
set user password info="details"
so it may be possible to write a script to run that for all of the users out of your current user database or from a user list.

See the following sections in the manual for more details:
External Authentication
LDAP External Authentication
NWAuth External Authentication

14. **How can I have some users who can connect direct to DPOP but others who can only connect with DMailWeb/CWMail?**

Q:I want to have two different types of users. I want one group to have both pop and web access to their mail, and I want the other group to have web access only. How would I set this up? Would I need to run two seperate servers? I plan to authenticate using an external authentication module (talking to a MS SQL 6.5 database).

A:Yes, you can run two separate servers or you can make an external authentication module flag some users as being only allowed web access.

The trick is that DPOP only has the ip_address that the user connected from to know if the user has connected from CWMail or with another email client direct to the POP server. DPOP passes this ipaddress to the external authentication module.

So,
1. If you run two separate servers then you can use the user_ip_address setting on one of the servers to only allow connections to that server from the ip address of the cwmail machine. Each server then either needs its own authentication database or you need an external authentication routine for each server which cannot 'see' the other server's group of users in the database.

2. The nicer way is to make your user database have a flag for each user to say whether they are allowed to connect directly to the POP server or not, and then make your external authentication routine check this flag, and reject the connection if they have not connected

from the appropriate IP address. The IP address that the user connects from is given in the authentication request by DPOP, e.g.
check username password ipaddress

So your authentication routine needs to check the "direct dpop connection allowed" flag and if it is false, it should check the ipaddress passed against your CWMail server(s)'s ip address and only allow the connection if it does not match. This is an example - you do not necessarily have to do it this way. The fact that the connection from IP address is passed to the external authentication module is the important point.

If I have not pointed it out before we also have the source code to another customer's SQL authentication module which I can give to you if it would help.

For more information contact
support- dmail@netwinsite.com

15. **How can I check what aliases I have set up for a user?**

Q:If I send a message to user x, how can I check what aliases are set up for that user?

A:To do this you should send a message to that username and then check the log file for lines with the word "chain" in them to see where it has been forwarded to.

You need to set,
log_chain true
in dmail.conf and then issue the command,
tellsmtp reload

You probably don't want to bother the user with a message, so you should make use of the tellsmtp command,
tellsmtp scriptfile.msc
to initiate a message to the user, but pull out before sending any data.

E.g. here is a scriptfile, bob.msc, that does this for a user bob
****
HELO domain.com
Mail From: <test@domain.com>
Rcpt To: <bob@domain.com>
QUIT
****

Once you have run the tellsmtp script (on debug log_level), then you can 'grep' or 'find' for lines with the word, 'chain' in the log file, dsmtp.log.

The following is a transcript of such an operation - looking for aliases and forward rules for the user bob.

C:\dmail>tellsmtp bob.msc
220 domain.com DSMTP ESMTP Server v2.5d

Send (HELO domain.com)
250 domain.com. Hello domain.com (161.29.99.1)
Send (Mail From: <test@domain.com>)
250 Command MAIL OK
Send (Rcpt To: <bob@domain.com>)
251 Command RCPT OK
Send (QUIT)
221 Command QUIT domain.com Service closing transmission channel to domain.com
Send (QUIT)

C:\dmail\log>find "chain" dsmtp.log

---------- DSMTP.LOG
26/04 11:53:40 *** Starting rcpt chain for bob
26/04 11:53:40 *** Adding <|\dmail\drespond.exe \message.txt -subject whatever -from "root@domain.com"> to rcpt chain
26/04 11:53:41*** Adding bob to rcpt chain

Which shows that the message is delivered to the robot '\dmail\drespond.exe . . .' and to the user, 'bob'

Note: The log lines with the word 'chain' in them were only added, in version 2.5d, so if you are using a version of DSMTP older than that then you will need to grep for something like, 'process' and work a bit harder to interpret the results :-)

---

16. I'm getting a Read Failed 109 error message, what's that?

Q:Dpop.log is showing the error message 'Read Failed: 109', what's that?

A:The 109 error says that a "pipe has broken". The two things in dpop that use pipes are external authentication processes and dslave processes.

Most likely it is the external authentication process causing the problem, and it is probably occurring on the read that DPOP does after sending the 'exit' command to the external authentication. I.e., DPOP has told the external authentication to quit but does not get a response from it. So it checks to see if the external authentication has responded every so often (you will see the 109 error in the log every time that it does) until the timeout period is reached and DPOP gives up.

So this suggests that the external authentication routine is either not returning,
+OK\n
(+OK with a carriage return at the end) when it receives the exit command, or that it does not flush the output.

NWAuth has at times done both of these things. So you should probably upgrade NWAuth to a version from the 2.5d or higher distribution set (NWAuth 2.0b).

Note: To upgrade just nwauth you need to copy the nwauth executable file over your old nwauth file, e.g. on NT, \dmail\nwauth.exe. You will need to stop DPOP and DSMTP first

so that they stop all their NWAuth processes.

If you have your own authentication module then you should check that it does both of these things. Contact support- dmail@netwinsite.com if you have questions or a problem with this.

The other possibility for the error is that one of the dslave processes is no longer alive when DPOP thinks that it should be. If you do a tellpop status command it will show the number of slave channels that it thinks are running.

If this happens just once then it is probably not a problem, but if it continues to happen then it obviously does become a problem.

If the slave_number setting is above 0 then DPOP should always be running at least one slave process. Versions of DPOP before 2.5g had a problem with the dslave processes finding the dmail.conf configuration file, so if you cannot start a dslave process from the command line then this may be the problem. It will be evident in the log file, dslave.log (which itself may be being written to a strange directory on your machine - it is best to use a search to find it).

---

17. Can I filter messages based on the attachment name?

    There is no direct setting to filter by attachment filenames, but I believe that it can be done!.

    In the manual on our site(link below) under common optional settings you can find a setting msg_filter < filename>

    This points to a file which you create as just plain text and into which you can enter very basic filtering rules.

    But let's say we wanted to filter emails with the attachment filename of 'happy99.exe'

    We could have

    msg_filter f:\dmail\filter.txt

    and in filter.txt

    reject body begin 0666 happy99.exe
    reject body Content-disposition: attachment; filename= "happy99.exe"

    These two rules should pick up the required messages. The first reject rule is for uuencoded attachments and the second rule is for the more common MIME encoded messages.

    The rejection rules are done on simple string searches, so we suggest that you send a test message with an attachment to yourself and open up the drop file in a text editor. From this you can identify for yourself this text within the body of such messages. You will then be able to refine your rules to catch the type of attachments your users get.

    You will no doubt find the command,
    tellsmtp filters
    useful as it lists all filters found,and their number which corresponds with the rule number

given in the line logged when a filter is matched by an incoming message.

NB: you cannot use wildcard characters in body filter rules!!!

reject body *.vbs
will not work, you should have,
reject body .vbs
in order to be a little less general we suggest,
reject body .vbs"

You can use wildcards in header processing filters - DSMTP uses a different sort of processing for them, because they are shorter and therefore do not need to be processed so efficiently.

There is another problem to the suggestion above. Sometimes an email client might split the, Content-disposition:...
line on to two lines. In which case the suggested filter will not pick it up.

The suggested filter above is still worth adding, but we are working on a MIME parser which extracts all the MIME details so that attachment filtering and other filtering will become much easier.

Please contact DMail Support for an update on when that will become available.

---

18. Tell me about the SMTP protocol?

The SMTP protocol is the way that an email client talks to an SMTP server in order to send a message. Note: Often it is two SMTP servers talking to each other (relaying), rather than an email client and a server.

A typical SMTP transaction looks like (this is NOT an RFC example),

client: (opens TCPIP connection to port 25)
server: 220 tosh.com DSMTP ESMTP Server v2.5f
client: EHLO tosh.com
server: 250-tosh.com. Hello tosh.com (161.29.2.46) < cr>
250-ETRN<cr>
250-DSN<cr>
250 HELP
client: MAIL FROM:<bob@tosh.com>
server: 250 Command MAIL OK
client: RCPT TO:<tam@tosh.com>
server: 250 Command RCPT User found OK
client: DATA
server: 354 Command DATA Start mail input; end with < CRLF>.<CRLF>
client: From: bob@tosh.com
client: To: tam@tosh.com
client: Subject: hello
client:

client: this is the message body, line 1
client: line 2
client: .
server: 250 Command DATA Processed mail data Ok
client: quit
(server drops TCPIP connection)

Notes:

- The client sends EHLO rather than HELO if it is capable of Extended SMTP (ESMTP) Protocol
- The server advertises all of its ESMTP capabilities if the client opened with EHLO
- In the DATA stage the client sends all of the message headers and then a blank line and then the message body. It sends a dot on a line by itself to indicate that it has finished.
- If the ESMTP client wants to send a message body line with just one dot on it then it should 'dot stuff' and send two dots and the DMail servers know how to handle this.
- If the client wants to be notified of the message delivery (not reading confirmation which is handled by the receiving email client) then it can specify a DSN. E.g. MAIL FROM:<bob@domain> NOTIFY=FAILURE

  Where FAILURE could be, NEVER, FAILURE, SUCCESS and/or DELAY. See Bounces and DSNs and also RFC1891

To send an email message without a client (and to enable you to try out SMTP protocol) you can create script files (filename.msc) for DSMTP and run them with tellsmtp.

Note: For the definite word on SMTP please search for the SMTP RFC on the internet (RFC821).

---

19. How do I add Multiple IP numbers on a single machine?

Windows NT: (workstation 4)

You need to edit the properties of your TCPIP Protocol to add the new ip address to your network card (NIC).

Go to the Network settings section of the Control Panel, select the Protocol Tab, and then select TCP/IP Protocol and click the Properties button.

You will be presented with the Microsoft TCP/IP Properties dialog window. On the IP Address tab, click on the Advanced button.

Select the network card (NIC) to which you wish to add the ip address. Then click on the Add button and enter the new IP address and the netmask for your network (if you don't know your netmask copy the one for the other ip address - a reasonable guess is 255.255.255.0).

Unix based platforms:

It is fairly easy to add multiple IP numbers for a single machine, up to 255 per interface is fairly straightforward. 1024 is usually possible with minor patches. The exact method varies from one form of Unix to another see http://www.nethelp.no/net/vif/readme.html for more information.

As an example on Linux you would do the following:

```
su - root
ifconfig eth0:2 999.59.4.31 up
```

to add a second ip number 999.59.4.31. The number :2 can be anything between :1 and :255

---

20. Can I specify a RANGE of IP addresses?

For most settings in dmail.conf that take an ip address, you can specify a comma separated list of entries (no spaces after the commas as a general rule) and you can also specify a range or wildcard.

We DO NOT gaurantee that you can use all of them for every setting, but we do try to code with this flexibility. So if you are wondering if a setting will take a range for example then try it out, don't just expect it to work :-)

NB: If a setting is a 'restrictive setting' then to get through the restriction a value must get through all the restrictions in the comma separated list.

Here are some examples:

NB: Some of the examples in this FAQ were incorrect. Fixed 23 May 2000.

NOTES:
'!' indicates NOT
'*' is a wildcard (generally for use at the start or end of a string, but with ipaddresses can be useful in the middle)
'?' is a single character/digit wildcard
'x-y' is a range from x to y (including x and y)

NB: you can use, '!*?' OR a range, you **can not** use both, so this is not allowed,
user_ip_address *,!1.1.1.0-255 (**bad**)

The examples use the setting user_ip_address which restricts what ip addresses can connect to dpop.

1. user_ip_address *,!161.29.5.24
allows all ip addresses to connect, except 161.29.5.24

2.
user_ip_address *,161.29.3-5.24
allows the following ip addresses to connect,
161.29.3.24
161.29.4.24
161.29.5.24

3.
user_ip_address *,!161.29.5.*
allows all ip addresses to connect, except,
161.29.5.0
...
161.29.5.255

4.
user_ip_address 161.29.3-5.0-255
allows the following ip addresses to connect,
161.29.3.0-255
161.29.4.0-255
161.29.5.0-255

5.
user_ip_address *,!161.29.*.24
allows all ip addresses to connect, except,
161.29.0.24
161.29.1.24
161.29.2.24
...
161.29.255.24

6.
user_ip_address *,!161.29.20?.24
allows all ip addresses to connect, except,
161.29.200.24
161.29.201.24
161.29.202.24
...
161.29.209.24

Note with this last example, if an ip address was, 161.29.009.24 then it would be allowed to connect.

---

21. I want to UPGRADE, ... ?

An upgrade is in general a quick and simple procedure. The same utility that you used to install DMail, dmsetup, has an upgrade option that does it all for you.

Note: we are always very careful when making changes to our programs that we do not 'break' them for existing setups. Having said that it is an easy thing to do so upgrading is not something we recommend doing whenever you feel like it - "don't fix what isn't broken" if you like. You should take particular care when upgrading from a version that is much older than the current beta version (e.g. 6-12 months).

Things to consider when upgrading the DMail server (or a part of it):

1. See the updates page,

http://www.netwinsite.com/dmail/updates.htm
to see which version you wish to upgrade to. If you are not sure then contact dmail support to confirm the version you should upgrade to. This applies particularly to versions out of the beta directory of the FTP site, ftp://ftp.netwinsite.com/pub/dmail/beta

Note: you can if you wish only upgrade one of the servers or utilities from the dmail distribution set - if you are after a particular feature in a recent beta release then this is often a good option.

2. Download the distribution set from our ftp site, ftp://ftp.netwinsite.com/pub/dmail

   If you are ftping from a command line then login as the user 'anonymous' and provide your email address as a password, then cd to pub/dmail.
3. Save a copy of your configuration file, dmail.conf (typically \winnt\system32\dmail.conf or /etc/dmail.conf)
4. You may want to revert back to your current version, so just in case you should try to save a copy of each of the executables that your system uses. If you have your last distribtion set then that should be enough. If not then you should save each of the server directories, e.g. \dmail (typically contains dpop, dsmtp), \dmail\dwatch, \dmail\dlist.

   DMSetup will not touch any of your critical data.

   For Your Information ...
   The critical data for your email server is almost all in the mail drop file and bin file directories, (defaults are, \dmail\in and /var/mail). The upgrade will not touch these directories, but of course if you wish to back them up then that is never a bad idea.

   The other critical information to think about is:
   a) mailing list information (lists.dat and users.dat for each list) - stored in the dlist directory which should be fairly small to back up.
   b) If you run external authentication then your user data base may be in a directory which dmsetup works in. NWAuth stores the user database in the dmail directory in nwauth.txt and on newer versions in nwauth.add as well.
5. Shutdown the DMAdmin windows GUI tool if you have it open (dmsetup can't upgrade dmadmin.exe if it is running).
6. Unpack the distribution set and run the utility dmsetup.
7. DMSetup should detect that you already have DMail installed and offer the upgrade option (2). DMSetup will stop each of the servers and then copy the new versions of the executables over the old ones. It will also upgrade your manual pages, *.htm in the dmail directory. Once it has finished upgrading it will ask you if you want it to start the servers again.
8. You should now check that the new version is working. You should at least,
   a) send a message through the system and,

b) if you use dlist, post a message to a mailing list.

If you suspect that something has not upgraded, then you should attempt to manually stop that server or program and then run dmsetup again.

If you have problems then please do contact dmail support .

---

22. I want to MOVE DMail, ... ?

Moving DMail to another machine is a fairly easy procedure. Here is a suggested method to help you remember the most common things. Each setup will be different so think if there are any other things that you need to copy over for your setup.

Note on License Keys:

> Your DMail license key was created for your old machine's specific machine name, e.g. server1.your_domain.com (UNIXish machines) or SERVER1 (Windows machines).
>
> If the new machine has the same name as your old one then simply load your key into the new machine with the tellpop command,
> tellpop key xxxx-xxxx-xxxx-xxxx-xxxx
> at the point below where you have started DPOP.
>
> If the new machine has a different name, then you need to email our Sales department,
> sales@netwinsite.com for a replacement key. You need to tell them the name of your new machine. They should email you your new key within 48 hours (usually only 24 hours).
>
> If you don't yet have your new key, do not worry, when when you start dsmtp it will create itself a temporary trial period key. So it should start and work straight away for you.

Suggest Method for Moving DMail ...

1. install the same version of dmail on the new machine but don't start the server when the installation utility asks you if you want the servers started
2. copy across to the new machine your dmail.conf file typically /etc/dmail.conf or \winnt\system32\dmail.conf
3. Copy over any other files included into dmail.conf or referenced in it, e.g. alias files.
4. Edit your host_domain settings in dmail.conf (and your dpop_host setting) so that your new machine name is included **at the end** of the list of host_domains (also known as synomyms)
5. now if it won't impact on your old server, start the new server up and try sending a few test messages through it

Once you are ready to switch completely to the new machine ...

6. Stop all servers on both machines

7. Copy over the mail drop files, e.g. /var/spool/mail or \dmail\in

   NB: if your bin_files and _inf files are in other locations don't forget to copy those as well.
8. Copy over the work_path directory, e.g. /usr/local/dmail/work or \dmail\work
9. Check dmail.conf on the new machine to see that all directory paths exist and that you have copied over any necessary things
10. Start up the new server and monitor it for the next few hours.

If you have problems then please do contact dmail support .

---

23. **I want to park mail for a domain (but mail is rejected as no relaying)**

    The setting that you need is,
    relay_to etrn_domain
    so that DSMTP will always accept mail destined for the domain etrn_domain.

    Then dsmtp will accept the mail and park it when it cannot connect to the server.

    It will try to send it every 2 hours and bounce it after max_retrytime hours (default is 2 days).

    When the connecting email server sends the ETRN command dsmtp will try to send all mail addressed to that domain in its queue.

    The other setting that you can use to bypass the DNS record if you have problems is,
    gateway etrn_domain ipaddress
    so that dsmtp uses the ipaddress given rather than doing a dns lookup on etrn_domain.

    In versions 2.8e and above, we added a new setting to DSMTP for that can also help with this. It is suspend_domain, e.g.,
    suspend_domain fred.com
    This setting stops DSMTP from processing any queue files destined for this domain, unless specifically requested by an ETRN commmand. So it is a good setting to use if someone will not be collecting their mail for a period of time longer than max_retrytime. NB: it can also be a bit dangerous to use for that same reason.

    In 2.8e we also added the setting, etrn_relay which allows all servers in a server farm or load sharing arrangement to receive an ETRN command sent to just one server.

24. **Can I run DSMTP (and DPOP) on another port?**

    Yes, the setting that you want is,
    smtp_port 1025
    then restart dsmtp (with DMAdmin or on UNIX platforms with,
    tellsmtp shutdown
    /usr/local/dmail/dm_start.sh
)

Similarly for dpop,
pop_port 1110
(/usr/local/dmail/dpop_start.sh to start dpop on UNIX).

NB if you are using dmadmin then you will have to select a new host to monitor with the following syntax as the ip address,
127.0.0.1:1025:1110:
so that it looks for the servers on the correct ports.
(you may need to set the password for this to work, with,
tellpop pass xxxx
,where xxxx is the password)

25. **Can I delete queue files from the queue?**

Yes, you can delete or move them with the result that that message is not delivered, however there is a big BUT...

Currently if you move queue files out of the work directory (work_path) you cannot easily put them back in. You can copy a queue file back into the work_path directory and dsmtp will pick up on it the next time it reaches that queue file number. But dsmtp may have created another queue file of that same number, so if you overwrite it then that message will be lost.

Also note that some queue files will be in use by dsmtp and so locked. The tellsmtp status command gives you information on what queue files are in use.

More information: See the section on Queue Files in the Disk Use and Files section.

---

26. **What things can I do to secure my mail system against hackers?**

Here is a list of things that we can think of. If anyone has suggestions or gets hit by a hacker please let us know so that we can add to this list.

- In general use ssh when sending root password across internet
- Use fake_vrfy, so that dsmtp responds falsely to checks on usernames on your system
- Use smtp_welcome (version 2.8a and above only) to hide what SMTP server you are using, and what version it is.
- Set manger_ip_address to limit manager commands to coming from as small a number of ip addresses as possible
- Use the tellpop password command to set your manager password to something secure
- Use shadow password files, which dmail supports when authent_method is set to unix_user (linux users use libc6 download).
- Check what UID your 'robots' run as, see Robots running as root - **Security Note**
- If a hacker is trying to guess passwords you will see a lot of the following messages in dpop.log on info log_level,
  Info: Rejected bob, authent said bob password wrong or not a valid user
  So you can search for the keyword, 'Rejected' in dpop.log

27. **Does CWmail and DMail server support multi-threading?**

    Yes and No. I will explain.

    First DMail:

    DMail is made up of an SMTP and a POP server, DSMTP and DPOP. Both of these servers are mostly just a single process and thread, so they would only run on one processor at one time.

    They have been written to be extremely efficient, and we believe that these servers are more efficient because of their single process architecture.

    However there are two 'bottle necks' for single process mail servers. To overcome these both servers can spawn subprocesses. Both DSMTP and DPOP spawn subprocesses for doing the user authentication, and DPOP also spawns a subprocess to 'burst' drop files, if a user's drop file is bigger than a certain size.

    So these subprocesses can be run on different processors to the main server processes.

    So Yes, DMail can take some advantage from a multiprocessor system, but it is not written as a threaded process.

    NB: it is worth noting that the biggest 'bottle neck' for an email server is the disk access times. Hence we recommend spending more money on fast disks rather than a multiprocessor environment.

    RE: CWMail

    CWMail is a CGI, as such CWMail runs as a single process spawned by the web server on practically every click on the web pages that it displays. So it depends on your choice of web server as to how worthwhile it is to run on a multiprocessor environment, but in general because each instance of the CGI running is a separate process in the OS environment, there should be no problem.

28. **Is there a limit to the length of a username?**

    Yes, there is. DPOP limits you to 78 characters in the username (this includes the domain name if you have set authent_domain true). So if your domain name was 10 cahracters in length, then you are limited to usernames of maximum length, of 78-1-l0 = 67 characters for local usernames.

    DSMTP does allow longer usernames because it needs to be able to relay on messages to people with longer usernames.

    NB: if you are using external authentication then the response that the module returns is not allowed to be longer than 1kbytes in total. So you will have to limit your length of username to something sensible, so that there is room to return long fwd="" fields for mail redirection.

    So if you impose your own limit of say 40 characters, you should not have any problems.

29. **Running DMail on your ISP's Server**

    We are often asked if it is possible to run DMail on an ISP's server.

    Basically the answer for DMail is no. The DMail server needs to be run with root privilege and in most cases a box can only run one Mail server.

    You can run DMail on your ISP's machines, if they are not already running a mail server on that box, or they provide you with a box at their site, for which you have root access.

    It may be an option for you to run a 'downstream' server on a local box of yours and have your ISP relay mail for your domain to you. DMail can send the ESMTP ETRN command to collect mail for such a domain.

    You may also be able to get your ISP to forward all your mail to just one POP mail account. Then the use of DMail's POPFetch is an option.

    Separate to the question of DMail is whether you can use one of our Web Based email CGIs such as CWMail on your ISP's 'virtual web server'. Please see the following FAQ for information on this,
    http://netwinsite.com/dmailweb/faqs.htm#Q18.

30. **Robots running as root - Security Note**

    Q:> We have customers who would like to forward e-mail into external programs.
    > However, we have had to disallow this because we noted
    > that Dmail was running these external programs as root.
    > How can we tell dmail not to run external programs as a priveledged user
    > and will this break auto-responders and mailing lists?

    A:If DSMTP can work out a user's uid (e.g. from the /etc/passwd file or from the authentication module response) then it will run the 'robot' as that user's uid.

    In the case of the question I think that our NWAuth authentication module is being used. It responds with lines like,
        +OK username config 0
    where the 0 on the end is the user's id. It returns 0, i.e. root, for ALL users.

    Also, up until version 2.8l if DSMTP could not work out a user's uid then it would run the robot as the same user as itself - i.e. root!

    This means that it is **important to restrict use of robots**, e.g. NetAuth only allows users to set the text of the autoresponder robot.

    On Windows machines it is not as common to allow access to users to create robots, but if it is allowed then the same issues need to be considered.

    Here are some options ...

    1. modify your authentication module to return a user id, e.g. that of the 'mail' user.

    2. We are adding setting,

robot_defaultuser <userid> <password - NT only>
which defaults to root if not defined.
If set then dsmtp overrides anything returned by the authent module so that all robots are run as the specified uid. If set to -1 then no robots are run. This should be available in 2.8l to be built 8 Jun 2000. It will apply to UNIX based **and** Windows platforms.

The DMSetup utility will add it by default on fresh installation in 2.8l onwards and prompt users to add it on upgrade.

You should specify a user with this setting that does not have any more privilege than it needs.

On UNIX platforms DMSetup will default this setting to the 'mail' uid, and you will probably want to create a special robot user with far less privilege. On Windows platforms DMSetup will set the setting to 'ROBOT_USR robot_usr' by default (i.e. username and password the same) and the sysadmin will need to create this account - probably in the Guest group.

3. Currently we have the domain_chroot setting, e.g.,
domain_chroot domainone.com /usr/local/robots
which makes all robots on the specified domain run with a root directory of, /usr/local/robots. I don't think that the robot can access outside of that with root access, but there may be clever trickery that hackers know.

4. you control what programs the users run via a web gui. E.g. drespond is an example of this. NetAuth controls who can run drespond and what options it is given.

RE: mailing lists and autoresponder

Mailing lists are not affected as DList handles these and is a separate process.

The Drespond robot is affected, but with all of the options above there is no reason why they cannot keep working. You may simply have to make copies of the executable in the domain_chroot directory etc.

31. Can I use DMail for a Remote or Dial Up Mail Server?

    Yes, DSMTP can be a remote or dial up mail server.

    Options:

    - DSMTP sending ETRN command to upstream Mail server (may be using RAS dialup):

      Setting the ras_timer makes DSMTP send the command, ETRN domainx.com, to the upstream server at the specified interval. DSMTP will send ETRN commands for all of your 'local' domains (as set by your host_domain or vdomain settings).

      The upstream server will then send all mail for those domains as soon as it can. Since your server is online it should be able to send the mail through to your local DMail server.

This is probaly the option to choose if you are retrieving mail for an entire domain or a number of domains.

See the links in the ETRN section for more information .

- Running POPFetch alongside local dsmtp for retrieving mail:

  POPFetch runs on the local mail server machine. It will periodically dial up your upstream server and collect all mail waiting in specified POP accounts. It will then process those messages and separate them out for individual users on your domain. It will feed the messages to the local DSMTP server so that it can deliver them locally.

  Often you can get whoever is running your upstream server to collate all mail for you into one POP mailbox for POPFetch to retrieve, e.g. in DSMTP this is easily done with the dmail.conf setting,
  forward *@yourdomain bob@domainx.com

  Follow this link for more information on POPFetch.

Note on Dynamic IP addresses:

If the machine where you want to run the Mail server does not have a Static IP address then you are probably limited to using POPFetch.

Some ISPs can support receiving an ETRN command for your domain when you are on a Dynamic IP address. It is not typical that they can as it requires specific dynamic DNS support,so you cannot infere that they are a sub-standard ISP for not offering it:-)

Note on bounces:

Using ETRN is a better option than popfetch if it is important that people sending mail to your local accounts receive 'bounce messages'. Most mail servers will try to deliver mail every few hours for a specified period if they cannot reach the final destination (your server) on the first go. At the end of that period, typically 1-2 days, they will 'bounce' the message back to the sender. With POPFetch (and some ETRN setups) the upstream mail server will consider the mail delivered once it recieves it (because it wrote the mail to a POP account). So if your server does not collect the mail for a long time (and nobody notices) then the sender would not be notified. ETRN can suffer from the same problem - so you should check with the upstream provider if it is a worry to you.

32. Can I use DMail from behind a firewall or proxy server?

    In most circumstances yes, but there are some circumstances where you may need to rely on an 'outside world' SMTP server.

    NB: we are using the term 'firewall' loosely. We will mostly talk as if you are running a Proxy Server on your firewall box, rather than a router.

    There are two main things that you need to provide,

    1. DSMTP needs some way to connect to a DNS server to resolve domain names to IP addresses.

2. DSMTP needs some way to connect directly to the outside world SMTP servers for non-local mail delivery.

Here are some options, (Option 4 will soon be our recommended solution)

1. **Run DMail on the firewall box itself (so not really behind the proxy at all)**

   For some firewalls you won't be compromising security greatly to run the proxy server on the firewall box so that mail bypasses the proxy. In most cases if doing this you would store all mail on the firewall box until it was collected by the local email clients. You could store the mail on a network drive if you had a file server for example, but in most cases you would probably not do this because setting up the network drive connection would lessen the security of the firewall box.

2. **Relay via a DSMTP Server on your firewall box (bypass the proxy server)**

   The idea here is that the two DSMTP servers, one on the firewall box lets call it A, and one behind the firewall box (B), can pass on to each other the messages that each can not deal with. In this way the DSMTP server on the firewall allows mail to bypass the proxy server but no mail is stored on the firewall box.

   Outgoing mail will be 'gatewayed' from B to the firewall DSMTP server A which has access to the non-local SMTP servers and the DNS server(s) for non-local mail delivery. So A 'relays' mail for B.

   Incoming mail will arrive at DSMTP server A which will 'gateway' all local mail to DSMTP server B.

   To do this you need to,

   1. Tell server B to gateway ALL outgoing mail to server A
   2. Tell the firewall server A to accept outgoing mail for 'relay' from server B
   3. Tell the firewall server A to accept incoming mail addressed to local domains on B
   4. Tell the firewall server A to gateway incoming mail addressed to 'local domains' on to B

   So if a.a.a.a is the ip address of server A and b.b.b.b is the ip address of server B...

   On server B add to dmail.conf,
   gateway * a.a.a.a

   On server A add to dmail.conf,
   forward_from_ip b.b.b.b
   relay_to domain1.com
   relay_to domain2.com
   gateway domain1.com b.b.b.b
   gateway domain2.com b.b.b.b
   (keep adding relay_to and gateway settings for all local domains)

   See also, Routing.

3. **Gateway all outgoing mail to an Outside world SMTP server (via the proxy server)**

   You can avoid most problems by **'gatewaying'** all outgoing mail to an SMTP server in the outside world, that provides you with 'relay' access.

   This is similar to the option above in that outgoing mail is relayed via an SMTP server with 'outside world access', but with this option, mail goes through the proxy server, and incoming mail comes **direct** to your proxy server.

   To do this you add a setting to dmail.conf like,
   gateway * x.x.x.x
   where x.x.x.x is the ip address of your firewall server.

   The possible problem with this is that you need to set up the proxy so that,

   A. anything connecting to port 25 from the DMail server address is mapped to port 25 at your ISP's SMTP server IP address.

   B. anything connecting to port 25 from other addresses (e.g. outside world ones) is mapped to port 25 on your DMail server's IP address.

   Some proxy servers are not capable of this type of setup on the single port (25), and some will do it 'automatically' with a 'SMTP proxy' feature. If you are using a router then it will probably have no problems with this.

   If your proxy cannot do that sort of setup, then note that in version 2.8n we have altered the gateway setting so that you can specify the port on the proxy,
   gateway * x.x.x.x:1025
   This allows you set up up two port mappings on the proxy,
   1025 -> ISP_IP_Address:25 (for outgoing mail)
   25 -> DMail_IP_Address:25 (for incoming mail)

   You also **must** get whoever is running the outside world server to accept mail from your server for relaying. ISPs by default will stop you from relaying through their box unless you have their permission (it is to stop them being abused by spammers). They will probably do this based on the ip address of your proxy server - as that is the address that mail from your DSMTP server will appear to them to have originated from. If they are running DSMTP then they would add the forward_from_ip setting for your ip address.

4. **Proxy DNS Access AND use telnet proxy to reach non-local SMTP servers**

   Sometimes people have their own DNS server behind or on the firewall, but for most people they don't so you have to,

   Set up a proxy server to relay all DNS lookups:

   Doing this varies between proxy servers. It is important to note that DNS lookups can be done on a TCPIP port and/or a UDP port. So you need to set up your proxy server to at least relay TCPIP connections on port 53 to port 53 on the DNS server. On most

proxy servers you can setup a TCPIP 'port mapping' or 'link' to do this.

You also need to tell DSMTP which DNS server to use by adding the dmail.conf setting,
dns_host y.y.y.y
where y.y.y.y is the ip address of the DNS server to use. You **must** restart DSMTP after changing or adding this setting.

Using telnet proxy to reach non-local SMTP server:

You cannot simply add a 'port mapping' for port 25 on most proxy servers and expect them to 'proxy' all incoming and outgoing connections on port 25 to/from the DSMTP server.

When the DSMTP server tries to reach a non-local server it is trying to connect to that server directly on port 25. Even if we added a setting to DSMTP to make it connect to your proxy server, there is no way for the proxy server to map an incoming connection on port 25 to the required server which could be anywhere in the world!

So we have recently added a new setting to DSMTP (in version 2.8n) which makes it open all non-local connections via your proxy server's telnet port.

Because there is no fixed syntax for proxy telnet ports the new setting allows you to specify the connection string to be given to the telnet server, e.g.
destination_ip:25

The setting is,

proxy_domain <wildcard_domain_name> ip[:<port>] <proxy_request_string [optional macro $IP]>

where $IP is the resolved IP address of the destination domain, E.g.,
proxy_domain * 1.2.3.4:23 $IP:25
where 1.2.3.4 is the ip address of your proxy server. This example results in all outgoing mail being sent to the telnet proxy at 1.2.3.4, where the proxy server takes a request string of, x.x.x.x:25. DSMTP will replace x.x.x.x with the DNS resolved IP Address of the the destination domain.

5. ---

Does DMail support CDONTS?

No, but there is now an option in DMail to deliver mesages written to file.

I am afraid that CDONTS were created too much as part of the web server/email server combination and do not use the standard SMTP protocol that they 'should' for sending mail. So as far as I know there is no way for CDONTS emailing calls to get the mail message to the SMTP server.

However it would seem that it is an option (possibly the default) for CDONTS calls to write email messages to a given directory.

We have recently added a feature whereby DSMTP will 'pick up' messages written to file in a directory and deliver them to the destination address specified in the message headers in the file.

So given that you can somehow make your system create such files on the server's local drives, then dsmtp can deliver them.

For information on the setting needed and the message file format see the DSMTP Settings List,
spool_dir.

NB: you need a 2.8 version of DSMTP so I suggest that you download the latest 2.8 build (probably 2.8v) from the directory,
ftp://ftp.netwinsite.com/pub/dmail

NB: This new feature has not been thoroughly tested yet and we can not be sure that it will handle the file format created by CDONTS. So contact DMail Support if you strike any problems or need us to make changes to the system.

6. ---

My Users are not appearing in the nwauth database file?

Often people are mistaken about the way that nwauth stores usernames and other data, so here is an explanation.

When you add a user to nwauth, e.g. by running it at the command line

nwauth
set bob secret
quit

then nwauth will write the username and the details to the file,
nwauth.add
in this format,
username:password:blah
where 'blah' is any other information you store for the user.

When you modify a user's details, nwauth simply adds another line for the same user to nwauth.add with the new password or other details.

When you delete a user, nwauth adds a line like,
username:(DELETE):(DELETE)
to the nwauth.add file.

When the nwauth.add file reaches a certain size nwauth will delete that file and update the main database file, nwauth.txt. When it updates nwauth.txt it processes it in order, so in general it uses the last entry for a user found in the nwauth.add file and deletes the user if it finds a line for a user with the '(DELETE)' password. It does this so that all of its operations are instantaneous no matter what size the user database is.

Often you will only have an nwauth.add file, and the nwauth.txt file will not appear

for several days.

If usernames are not being added to the file here are some helpful hints:

- Look in the nwauth.add file not the nwauth.txt file
- Try nwauth from the command line. See EAP definition for details of the commands.

  If it works from the command line, then you probably have the incorrect setting in dmail.conf or netauth.ini. This is now, authent_process for both dmail.conf and netauth.ini. (On NT use a drive letter or UNC name when specifying the process, e.g. c:\dmail or \\machineA\cdrive\dmail rather than just, \dmail which is ambiguous).

  If it still fails then see the next suggestion below.
- **Is nwauth modifying the nwauth file in the directory you think?**
  This might be the problem if you are running nwauth across a network or on an NFS drive.

  If you are suspicious of this then search your machine for any copies of nwauth.txt or nwauth.add.

  NWAuth decides where to find/create the nwauth.add and nwauth.txt files in one of two ways.

  1. It looks at the **local** dmail.conf file and uses the value of the dsmtp_path setting, typcially c:\dmail\

  2. You run it with the command line argument, -path, to specify the path to use, e.g. at the command prompt,
  c:\dmail\nwauth -path c:\dmail
  or in dmail.conf or netauth.ini,
  authent_process c:\dmail\nwauth -path c:\dmail

  NB: you should not need to set the path unless you are running in a server cluster. We don't recommend that you use the -path option unless you need to - i.e. be careful of using it as a quick fix without understanding why it is not working without it. Talk to DMail Support if you want help working out why it is not working.
- **There could be a file permission problem:**

  (See also, Authentication for DMail and NetAuth on Clustered machines and Network Drives)

  On NT:
  nwauth is spawned by dsmtp and dpop, which are spawned by dwatch service which is typically running as the 'System Account', so check that the directory that nwauth is running in and the nwauth files give full access to that user.

If using NetAuth, note that it is generally being run as a specific user by the web server. You to work out what the user is (typically IUSER_XXXX, where xxxx is you machine name). Then ensure that that user is created on the box and has the permissions needed to run nwauth and create/access the nwauth.add and nwauth.txt files in the dsmtp_path directory.

On UNIX:
nwauth is spawned by dsmtp and dpop, which may be spawned by the dwatch process. All of these will be running as root, so in general you should not get a problem. If you are running nwauth on an NFS then you will probably need to set root access on the file share so that these programs can access it.

During installation the NetAuth binary should have had its s bit (sticky bit) set. It's ownership should also have been set to the root user. This is so that the web server will always run it as root.

Unless the permissions are set as such then NetAuth will not be able to function properly.

So,
ls -l netauth.cgi
should show something like this,
-rwsrwsr-x root:root netauth.cgi

If not then set these permissions with the commands...
chown root:root netauth.cgi
chmod 6775 netauth.cgi

NB: with file permission problems, it is often a good idea to give all access to the user to get it working and then work backwards restricting the access to the level you are happy with.

7. ---

Authentication for DMail and NetAuth on Clustered machines and Network Drives
(AKA: Running NWAuth on a shared network drive)

Most of the following is for the authentication module NWAuth, but much of it applies when using any authentication module.

When you have a cluster of DMail servers or a DMail server and NetAuth running on a web server you need to allow them to all access the same user database.

For authent modules like, MySQLAuth this is not a problem because the database is accessible via TCPIP from any machine on the network.

For nwauth and some other modules which use local database files this is a problem.

Here are 3 solutions for nwauth:

1. make all of the servers run the same copy of nwauth on a shared network drive.

2. run a separate nwauth on each server, and set the -path option so that they all work on the same nwauth.add and nwauth.txt files on a shared network drive.

3. run a TCPIP daemon that spawns nwauth on one machine and then run a 'client' for that daemon on each of the servers.

Option 3 has some good benefits, so we are creating a new module called, TCPAuth (with TCPAuth_client) to do that. Contact DMail Support for information.

Option 1 is the current option being used by customers so is known to work on UNIX and NT. Setup for option 1 is described below.

Option 2 is pretty similar to option 1, so if you want to do that read the suggestions below and you will probably be able to work out what to do.

So to recap, the information below is how to,

Run nwauth on a shared network drive.

- **For those on UNIX and using NFS drives:**

  nwauth is spawned by dsmtp and dpop, which may be spawned by the dwatch process. All of these will be running as root, so in general you should not get a problem.

  During installation the NetAuth binary should have had its s bit (sticky bit) set. It's ownership should also have been set to the root user. This is so that the web server will always run it as root.

  Unless the permissions are set as such then NetAuth will not be able to function properly.

  So,
  ls -l netauth.cgi
  should show something like this,
  -rwsrwsr-x root:root netauth.cgi

  If not then set these permissions with the commands...
  chown root:root netauth.cgi
  chmod 6775 netauth.cgi

  You will probably need to set root access on the file share so that these programs can access it.

  In both dmail.conf and netauth.ini use the authent_process setting to specify the full path to the nwauth process and pass it the command line argument, -path, e.g.
  authent_process /shared/dmail/nwauth -path /shared/dmail/
  (in dmail.conf the authent_method setting should also be set to,
  'authent_method external')

Remember to restart both DSMTP and DPOP after changing the authent_process setting,

tellpop shutdown
tellsmtp shutdown
/usr/local/dmail/dm_start.sh
/usr/local/dmail/dpop_start.sh

If authentication fails, then look in the dpop.log file to see why. You will see at the start of the dpop.log file after restarting dpop if it has had difficulty spawning the authentication process.

- **For those on NT and using network drives:**
  1. Run the dwatch service as a specific user, e.g. IUSER_DMAIL, which you must create on ALL boxes, i.e. the mail server box, the web server box and the box that holds the network drive (it will depend on your setup how many boxes this is, it may be just 2 boxes or many more).

     Set this in Control Panel, Services. Select 'dwatch monitor for dmail servers' and click on Startup and then change the check the 'Log on as this account:' button and enter the account (IUSER_DMAIL) to be used and any details.

     You will have to stop and restart the dwatch service in the Services dialog to make this change take effect.
  2. Similary you have to ensure that the Web Server spawns NetAuth as the same user, IUSER_DMAIL, so that it can access nwauth on the network drive.

     Most web servers allow you to set the username used for spawning CGIs (that is what NetAuth is). Often they are spawned as the anonymous user login account, IUSER_XXXX where XXXX is your machine name - look in your NT system user database for such a user.

     You won't know what the password for that user is, so you won't be able to add that user to the other boxes in your cluster. This is why we suggest creating the new user, IUSER_DMAIL, on all of the boxes.

     If you have the IIS server see the specific note below.
  3. Use UNC names for the paths rather than mapped network drives, e.g., authent_process \\machineA\Cdrive\dmail\nwauth.exe

     UNC names allow the dwatch service which will start automatically after a reboot to reach nwauth on the other box even if no one is logged in yet. Whereas mapped drives are only accessible once someone has logged in to the box, so won't be accessible to dwatch (and hence dsmtp and dpop) after a reboot until someone logs in to the mail server box.

  4. In both dmail.conf (c:\winnt\system32\dmail.conf) and netauth.ini

(c:\inetpub\scripts\netauth.ini) use the authent_process setting to specify the full path to the nwauth.exe file and pass it the command line argument, -path, e.g.
authent_process \\machineA\Cdrive\dmail\nwauth.exe -path \\machineA\Cdrive\dmail\
(in dmail.conf the authent_method setting should also be set to, 'authent_method external')

Remember to restart both DSMTP and DPOP after changing the authent_process setting. The best way to do this is either with DMAdmin or using the Control Panel Services dialog.

If authentication fails, then look in the dpop.log file to see why. You will see at the start of the dpop.log file after restarting dpop if it has had difficulty spawning the authentication process.

- **Special note on the IIS web server:**

  Follow all the suggestions above. If they do not work check the following magic setting as this sysadmin did:

  I just tried changing the settings in IIS.
  Under Web Site properties->Directory Security->Anonymous Access...->Allow
  Anonymous Access[edit]

  I have "IUSER_DMAIL" as the username and have set up all permissions for that user on both mail server boxes. I had ticked,
  'Enable Automatic Password Synchronization'.
  I unticked this, and NOW IT WORKS!

Notes on updates to DMail, comprising the DSMTP, DPOP and DList mail servers

(in reverse order)

Jump to: current release version
Jump to: current recommended 'safe' beta version

NB: Have you pressed your 'reload' button to make sure you are not looking at a cached version of this page!

For a list of known bugs see,
Known Bugs in Current and Recent Versions of DMail

Use this link to get to the release directory of our FTP site
Use this link to get to the beta directory of our FTP site

2.8z1 DSMTP BUG FIX: on solaris file handle problem when greater than 50 tcp_max setting, added SAVE_LOW flag as per dpop. - changed dotlock.c in dsmtp to log if normal file open occurs when get NFS permission denied error, and updated master dotlock.c, 8 Sept. 2.8y built Stu, Tues Sept 5. DSMTP and DPOP: updated dotlock code, so no longer fails every write on certain platforms, e.g. solaris sparc with permission denied error. (exclusive open not allowed on solaris NFS error). 2.8x built 4 Sept 2000, trw same as 2.8y as far as we know DSMTP and DPOP: updated dotlock code, so no longer fails every write on certain platforms, e.g. solaris sparc with permission denied error.

Version & Release Date	**Changes**
2.9b coming soon NB: this is the next line of beta versions after 2.8 Settings added/changed: log_copylines,smtp_auth_out Commands added/changed:	• **File Locking Bug on Solaris only:** now does not fail to secure lock on drop files on solaris sparc OS when using lock_id and NFS drive. Used to give 'Permission Denied' error when trying to lock drop file. May need to set, use_flock false, to turn off additional flock calls which will fail on NFS with permission denied error. Fixed in 2.9b, 2.8z2 • DSMTP BUG: fixed unreported slow reload bug on systems with large numbers of vdomains • DSMTP Fix: Will no longer pass low-ASCII chars to auth requests • DSMTP Fix: Will now disregard *any* out of sync authent replies, not just 'lagged' ones. • DSMTP Fix: authent processes that timesout 6 times in a row are killed, and respawned • **DSMTP and Authent Processes with command line arguments:** Previously only the first authentication process spawned by DSMTP, was spawned with the command line arguments. If you run a system where in your config file you specify command line arguments on the authent_process setting, e.g.,

authent_process c:\dmail\nwauth.exe -drop_path 2 c:\dmail\in
then you should work out if those options affect the authent processes's response. If they do then you should upgrade urgently to version 2.8z3. This is fixed in versions 2.8z3 and 2.9b

- DPOP Security Fix: now has stronger manager password guessing protection.
- --------
- DSMTP: Added support for MAIL SIZE extenstion (RFC1870)
- DSMTP: Should now log any errors during log rotation (after the fact, of course :).
- DPOP: Appends message to users drop file notifying him/her of message expiry when using expire commands.
- DSMTP: Added new setting, log_copylines <filename> <string>. No wildcards. Obeys the same size, number and rotation rules as dsmtp.log. This setting allows you to create your own log files that only log specific log messages, e.g. if string were 'error' then the file specified would have any log lines with the word error in them logged to it.
  Up to 16 different filenames may contain the $DATE macro, which gets replaced by ddmm (the current date) so that you end up with a new file for each day.
- DSMTP: added setting, smtp_auth_out <wildcard-domain> <username> <password> which makes dsmtp authenticate the smtp channel when talking to other (upstream) smtp servers for any messages where the destination domain is matched by the wildcard-domain setting. Use simply * for all domains, and you probably need a relay_to setting for the same domain.
  Note: the result of an AUTH transaction CANNOT cause DSMTP to retry/fail a message transaction.
- DSMTP: Made the qfile status info a bit more readable :)
- DPOP: Now logs expiry of messages in dslave.log in a nice way, e.g.:
  13 14:51:55 Info: expired for user0: Date: Wed, 13 Sep 2000 14:13:26 +1200 From: User0 Subject: Test
- DPOP FIX: Limited authent_number setting to no more than 10, and we still recommend just 5,

	as more than 10 does not give performance increase and slows dpop down. ● DPOP FEATURE: Added support for domain-specific bulletins. If the first line of a bulletin contains the header 'X-DPOP-Domainlist: ', then the bulletin will only be shown to users who are in the domains (space separated) in the header. The tellpop command now prompts for the domains, that each bulletin message should be sent to when you create it. If no such header then bulletin goes to all domains as before.
2.8z3 19 September 2000 NB: this is the current 'safe beta' hoped to become the release version in the next 2-3 weeks. Settings added/changed: Commands added/changed:	● **DSMTP and Authent Processes with command line arguments:** Previously only the first authentication process spawned by DSMTP, was spawned with the command line arguments. If you run a system where in your config file you specify command line arguments on the authent_process setting, e.g., authent_process c:\dmail\nwauth.exe -drop_path 2 c:\dmail\in then you should work out if those options affect the authent processes's response. If they do then you should upgrade urgently to version 2.8z3. This is fixed in versions 2.8z3 and 2.9b
2.9a 8 September 2000 NB: this is the next line of beta versions after 2.8 Settings added/changed: Commands added/changed:	● DSMTP BUG: dsmtp was trying to do NFS file locking whether lock_id set or not. It is now possible to turn it off. ● DSMTP BUG: NFS file locking caused large startup delays when restarting after process had died. Now much shorter (16 seconds per channel where lock file clearing has to occur) and will clear lock files immediately if older than two minutes old. ● ------
2.8z1 built 8 September 2000 NB: in release directory as will soon be release version Settings added/changed: Commands added/changed:	

2.8y to be built 31 Aug 2000 NB: in release directory as will soon be release version Settings added/changed: Commands added/changed:	
2.8x to be built 31 Aug 2000 NB: in release directory as will soon be release version Settings added/changed: Commands added/changed:	
2.8w to be built 31 Aug 2000 NB: in release directory as will soon be release version Settings added/changed: Commands added/changed:	• DList BUG: Fixed killer bug that posed minor security issue. • ------ • DList: Added lists.dat setting, max_users. syntax max_users ,Prevents new subscriptions to a given list, if the line 'max_users n' is present for that list in lists.dat, and if the current number of users in that list is greater than n. • DList: Added farewell message. If found, the farewell message is sent to a user when that user unsubscribes from a list. The filename used is 'leave.tpl', and is located in the same path as 'join.tpl', i.e. the lists directory.
2.8v 29 Aug 2000 NB: in release directory as will soon be release version Settings added/changed: Commands added/changed:	• DSMTP BUG: dsmtp was trying to do NFS file locking whether lock_id set or not. It is now possible to turn it off. • DSMTP BUG: NFS file locking caused large startup delays when restarting after process had died. Now much shorter (16 seconds per channel where lock file clearing has to occur) and will clear lock files immediately if older than two minutes old. • DSMTP Fix: add_footer is now case insensitive • ------ • DSMTP: Will now add a body to bodiless messages • DSMTP: Will now accept messages with invalid headers (unless told not to) • DSMTP: Will now start the message at the first invalid looking header, unless reject_badheader is true this means that messges sent without a separator line between headers and body will now have a valid body.

2.8u 23 Aug 2000 NB: in release directory as will soon be release version Settings added/changed: Commands added/changed:	• DSMTP BUG: Fixed bug where message_id-less messages would crash dsmtp. • ------
2.8t 22 Aug 2000 NB: not built for most platforms Settings added/changed: Commands added/changed:	• DSMTP BUG: 2.8s dies on some platforms when external authentication 'fwd' field has empty values, e.g., fwd="bob,,fred" • ------
2.8s 17 Aug 2000 NB: in release directory as will soon be release version Settings added/changed: Commands added/changed:	• DPOP BUG: fixed problem with dpop's channels not responding at regular intervals (all channels would stop responding, tellpop list_current shows channels in pre login stage). • DSMTP Fix: new SMTP AUTH feature was broken with respect to virtual domains. Previously dsmtp did not handle ip address based vdomains correctly (not putting on prefixes and authenticating against main domain when authent_domain true) in terms of SMTP AUTH authentication. • DSMTP BUG: Stopped DSMTP terminating its own process when delivering to with robot names that did not exist on UNIX based platforms • DSMTP BUG: in 2.8q we broke forwarding to /dev/NULL in forward settings or aliases, bounces messages, this is because of bug above where you can't pipe to something that does not exist. Now functionality of using /dev/NULL is not changed but you will get error messages everytime it is used because of new security checking code, so we recommend changing to using @NULL. • ------

2.8r
2000

NB: Not yet Released

Settings added/changed:
dns_switch_nfails,retry_invalid_domain

Commands added/changed:

- DSMTP BUG: Fixed hard loop bug, if all gateways point to local addresses, or if all host_domain entries have a matching vdomain entry, DSMTP would go into a hard loop. (ralph fixed dmc_removen, so ok with removing last entry)
- DSMTP BUG: Fixed silliness with .forward file addresses
- IMPORTANT SECURITY ISSUE, DPOP: further enhances blocking of cross channel password guessing by limiting by ip address failed attempts/hour.
- ------
- DSMTP: Now swallows X-UIDL headers.
- DSMTP: max_rcvd now defaults to 30
- DSMTP: added ini setting, dns_switch_nfails which sets the number of fails before switching dns servers. (set to -1 for switch on every fail)
- DSMTP: Fixed bad handling of "blahblah@x.com"@local.com addresses - this was causing some sites to fail ORBS and other anti relay site tests.
- DSMTP: Added setting 'retry_invalid_domain' (default false) to continue trying to deliver instead of failing
- DSMTP: Added alias_fallthrough (default true) command to set global alias checking after local failure.
- DPOP: add setting dpop_maildir so dpop keeps doing maildir as well as normal so that easy to move back from maildir.
  If set Maildir format drop files will be scanned by DPOP only, for use in converting from maildir. DLIST FIX: added 'rough matching' so that when checking for users, e.g. is this user a list member then for example, bob@mail.domainx.com will be matched if bob@domainx.com is in the list. (uncommented chris's earlier addition of this 31072000). DLIST: fixed handling of mime encoded subscribe messages. (Stu to test on ball). (copied across Chris's dlist.c). DSMTP: domain checks are now insensitive all: fixed all parts so that they work with domains that start with digits (changed any instances of if(isdigit(*name... to isip,) DWATCH: fixed bug where multiple copies could start due to port connection failing. DSMTP: add_footer command now has an insensitive domain parameter DSMTP: Max

	retry time now works with non-default retry intervals
2.8q 2000 NB: Not yet Released Settings added/changed: Commands added/changed:	• • ------
2.8p 2000 NB: Not yet Released Settings added/changed: Commands added/changed:	• • ------
2.8o 2000 NB: Not Released to beta Settings added/changed: Commands added/changed:	• • ------
2.8n 2000 NB: Not Released to beta Settings added/changed: Commands added/changed:	• • ------
2.8m 10 June 2000 NB: Use instead of 2.8l, Currently in beta/ask directory of ftp site ... Settings added/changed: Commands added/changed:	• DSMTP BUG: fixed bug where messages with '<From' in them caused dsmtp to die! • DSMTP FIX: drop_connection setting added earlier was not working, now does. • ------

2.8l Not released NB: This version will be 2.8j with Security Fixes Settings added/changed: Commands added/changed:	• DSMTP FIX: Fixed bug where the forward-rule rewriting code screwed robots up • DSMTP FIX: Fixed possible bug with nt_user authentication method (will probably only manifest on W2k systems, if at all) • DLIST: fixed multiple moderators bug. Now sends to first moderator on comma separated list. • ------ • DSMTP: Added robot_defaultuser <userid>, which defaults to root if not defined. If set then dsmtp overrides anything returned by the authent module so that all robots are run as the specified uid. If set to -1 then no robots are run. DMSetup will set as 'mail' id by default. See Robots running as root. • DPOP: adding setting, bulletin_hasheaders. When set true you can optionally give headers when entering bulletin, and DPOP will use them. You can have a mixture of bulletins that are just the message body and those that have headers.
2.8k 5 Jun 2000 NB: Special Security Fix 'Safe Beta' Version, based on 2.8i	• Fixed Exploitable ETRN command bug.
2.7r 5 Jun 2000 NB: Special Security Fix 'Safe Release' Version, based on 2.7q	• Fixed Exploitable ETRN command bug.
2.8j Never put in Beta Directory NB: Settings added/changed: external_processor Commands added/changed:	• DLIST Fix: fixed From address parsing so that addresses like, ' "user@domain" <user@domain>' do not get stripped to "user@domain" instead of user@domain. • DSMTP FIX: Fixed the NT bug with blank results from dns_host queries. So no longer dies on WIN 2000 when it finds a blank DNS entry. • -------- • DSMTP: Added 'external_processor' command. Default false. This is a beta system for passing mail to an external program. If set to true, DSMTP writes m_x.dmn when it receives the dot of the incoming message.DSMTP then Waits for m_x.rdy to appear. The first line in m_x.rdy should be an SMTP-style reply line. If it's error code >=400, DSMTP will use it, other it will process the message as per usual. (We will be writing a robot to test this soon - probably one that spawns common virus scanners)

	• DSMTP: added further Speed enhancements.
2.8i not released NB: This version has a number of speed enhancing changes in DSMTP. NB: New DSMTP DNS disk caching feature added by default in this version. Settings added/changed: dns_disk_disable, msg_filter_skip , msg_filter_skip_onlyif, forward_from ,forward + forward_cc,log_mime reject_case attachment Commands added/changed: tellsmtp offline	• DSMTP: fixed AUTH PLAIN bug, where stray characters from a previous AUTH transaction would be appended. Bug resulted in AUTH for a user failing until restart after several successfull logins. • ------- • DSMTP: Added command "hash_qfiles". Default false. Applies directory hashing for qfiles. This can give a significant speed increase if set to true on UNIX based platforms if the number of queue files gets above 2000. NB: Currently, dsmtp will put unhashed qfiles into the right place but can't unhash them. So you cannot just turn this setting on and off. • DSMTP: Speed Enhancement: Added internal tracking of qfiles to cut down on disk thrashing. • DSMTP: Speed Enhancement: Added internal tracking of qfile record count. • DSMTP: Tracked down rogue "PANIC, channel not initialised" messages • DSMTP: Added 'tellsmtp offline' to make DSMTP refuse all incoming connections (except from 127.0.0.1, need to use 'tellsmtp -o 127.0.0.1 online' to put it back online.) • DSMTP: now caches successfull dns lookups to disk, with setting dns_disk_disable to turn it off. • DLIST Improvement: send to rest of users on list when dsmtp tcp connection fails. It waits there forever trying to reconnect. • DSMTP: Added explicit logging of connection rejection by remote server. • DSMTP: Added msg_filter_skip <string> - wildcard match against address from rcpt to line, msg_filter_skip_onlyif <string> to trigger message filter bypassing, msg_filter_skip_onlyif bob (= deliver ONLY IF all recipients are exempted) msg_filter_skip bob (= deliver message if bob is the only OR one of the recipients, regardless of whether the other recipients are exempted or not) • DSMTP: Added vdomain and hostdomain keywords for forward_from • DSMTP: forward and forward_cc can now rewrite. E.g. the following syntax, forward *@test.com %1@domain.com results in all mail to the userx@test.com being redirected to userx@domain.com. (or %1@ to

use the original domain, e.g.
forward abuse@* postmaster@
would forward all mail to 'abuse' at any domain to 'postmaster' at the same domain)

- DSMTP: Added 'log_mime' command (defaults to false) to log stuff about MIME parsing.
- DSMTP: Added beta setting 'reject_case attachment <wildcard-string>' and 'reject_nocase attachment' for MIME filtering (not weighted, like the rest of message filters, so if hit will always result in rejection, cannot be bypassed with an accept msg_flter line).
  E.g. reject_case attachment *.vbs*
  will reject all attachments with '.vbs' in their names.
  NB:This setting does its best to find attachments in MIME encoded messages. We will be improving it over the coming weeks. If you have a message that gets through this filter and you think that it should not then please send it to DMail Support
- DSMTP: Added buffering code for outgoing TCP data.
- DSMTP: Added $user$ macro parsing for maildir systems.
  DSMTP: Fixed maildir directory setting.
  We are adding maildir for use on NFS drives to DPOP soon as an alternative to our lock_id code.
- DSMTP: Nailed memory leak when reloading message filters.
- DSMTP: Added 'reject_nocase', 'reject_case', 'accept_case' and 'accept_nocase'. These are added so that we can tidy up the use and documentation of message filters.
  Body commands were *not* case sensitive by default (therefore reject body == reject_nocase body)
  Header commands *were* (therefore reject header == reject_case header)
- DSMTP: Added MIME parsing code, plus reject based on the 'name' parameter to 'Content-Type:'
- DSMTP: Now takes notice of the dtablesize setting
- DSMTP: added setting,
  drop_connection <string>, which will cause dsmtp to drop any connection where found as part of the SMTP envelope. E.g.,
  drop_connection COOLSPAMMER

	will cause the TCPIP connection to be dropped if 'COOLSPAMMER' is found in any envelope command, i.e. MAIL FROM, RCPT TO, HELO etc.
2.8h not released NB: Settings added/changed: rbl_exempt,self_rcvd, delay_badrcpt (DList: log_bounce, bounce_remove) Commands added/changed: tellsmtp readraw	• DSMTP BUG: fixed a "panic ...channel not initialised" crash. • DSMTP BUG: fix of dns code that may have caused dsmtp to die occaisionally, due to it using corrupt data. • DSMTP BUG: Fixed bug where mail to a non-existent vdomain- passwd user would go an existing /etc/passwd user • DSMTP FIX: Fixed filter_file problem on reload - filters were being lost on tellsmtp reload. • ----- • DLIST: Added new setting, join_cookie_subject true , which modifies cookies to use the subject line. • DLIST: Added new setting, access_block bob*netwin.co.nz,fred@* ,which causes dlist to block all emails from these addresses. • DLIST: Added list setting for debugging LOG_BOUNCE true/false - if true then dlist will create a log file for the list of all bounces that it finds. It is intended that you use this setting for debugging only. • DLIST: Added new list setting, BOUNCE_REMOVE true , which enables auto removal of bounces from a list. The list of removed email addresses is emailed to the moderator at the end of each day and also written to the file, removed.log • DLIST:Added email command test_removed, which if received makes dlist send you the remove.log file. • DSMTP: Added delay_badrcpt command to send DSNs instead of 5xx replies for invalid RCPTs. With this setting set to true, DSMTP will accept all mail and send bounce messages if necessary rather than notifying the email client directly. We do **not** recommend this setting - we only added it to get around outlook wierd behaviour :-) • DSMTP: (added in earlier version) Added code to ensure that a reasonable number of channels remain for outgoing messages - half of max_send value.

	• DSMTP: Added beta tellsmtp command, tellsmtp readraw command to insert raw messages into q-files, e.g., tellsmtp readraw c:\dmail\oldmail*.msg would make tellsmtp try to convert all .msg files in that directory into queue file messages and then add them to dsmtp's queue in the work_path • DSMTP: Added self_rcvd setting to explicitly alter DSMTP's tolerance for talking to itself, this defaults to 15. So by default if dsmtp sees a message 15 times then it bounces it. • DSMTP: rbl_exempt, ban_ip, tar_except and forwardip now accept ranges, e.g. ban_ip 1.2.3.0-3, see config.htm#notes • DSMTP: Fixed ban_revname setting, so that it works • DSMTP: Added rbl_exempt setting for making ip numbers exempt from rbl actions
2.8g 27 April 2000 NB: Settings added/changed: lock_id Commands added/changed: tellsmtp testrcpt	• Fixed DSMTP Bug : Locking code fixed for NFS drives for DSMTP (DPOP still to be fixed - only causes problem if drop_users or tellpop drop is being used). • --------- • DSTMP: Some DSN messages have had their text changed. • DSMTP: Added tellsmtp testrcpt to track final result of user lookups.
2.8f 10 April 2000 NB: this version was removed from beta directory. Settings added/changed: stop_listen, qfile_split, smart_reload Commands added/changed:	• DPOP BUG: File Locking - DPOP was freeing a lock that it did not have, resulting in occaisional locking failure. Dotlock code changed so that this can not happen any longer. • DSMTP: Fixed tarpit channel leak. This appears to be a new bug, but I can't tell when it arrived • DSMTP: RSET no longer resets AUTH status - we backed the wrong one of the 2 conflicting RFCs :-) • DPOP: fixed bug where occasionally a burst (if it went through dslave) would result in user getting message, "error slave said open logfile ./log/\dslave.log" • DPOP FIX: list response was occasionally not sending closing dot when dpop logged that it did! • DSMTP: fixed bug in dns code -socket read error bug, particular brands of dns server that send data in small packets instead of whole

request in one go.

- ---------
- DPOP: added setting, stop_listen. Default FALSE. When set to true on shutdown, dpop will stop listening for new connections rather than rejecting them with 'error dpop offline'. This was added specifically for routers which will stop routing if dpop stops listening. So a nice shutdown is,
  tellpop shutdown 10
  which causes dpop to stop listening for new connections and give current connections 10 seconds to finish before shutting them.
- DSMTP: Added qfile_split command (default 1000, min 20, max 10000) to cap the number of RCPT lines per qfile. NB: this changes the way that dsmtp handles the que files for messages to a large number of recipients. Previously they could be slow to deliver because dsmtp deliers to the recipients in the que file pretty much synchronously. Now that qfiles like this get split they can be worked on separately.
- DSMTP: Made the spam_dump file into a proper dropfile. So you can point the setting at a drop file and then check the dumped mail with a normal email client through DPOP.
- DSMTP: now logs "panic ...channel not initialised" to help track sporadic crash.
- DSMTP: ORB/MAPS is bypassed when ip matches forward_from_ip or is recent_pop
- DSMTP: Added code to ensure that at least (max_send/2) chans are available for outgoing messages
- DSMTP: Now show ini_read error/warning messages in tellsmtp status
- DSMTP: Now gives message size in tellsmtp status
- DSMTP: Added SMTP AUTH stats (cached vs. non-cache) for 'lookup' and 'check' queries to stats_dump output file.
- DSMTP: authentication lookup cache raised from 1000 to 10,000.
- sysauth: put in regan's latest code fixes bug where sysauth returns Config resulting in dsmtp and dpop writing to file named 'config'. - 16 March 2000.
- DSMTP: Added logging of ini_read errors, even on success

	• DSMTP: Added smart_reload tolerance of missing #included files • tellpop: fix so that domains starting with a number would still be resolved before tellpop tries to connect, e.g. 200fred previously would not get resolved so connect would fail • DSMTP: Made DSMTP do a log rotation at startup rather than renaming dsmtp.log dsmtplog.old • DSMTP: Added 'smart_reload monitor' which rotates out the temp.conf files instead of deleting them
2.8e 23 February 2000 NB: Settings added/changed: lock_id,maps_action,authent_cache spam_dump(goes with spam_filter), dlist - footer and footer_html Commands added/changed:	• DPOP,DSMTP and IMAPD BUG: fixed all three so that file locking with lock_id does not get the occaisional error that it was getting. Lock_id setting should now be used on NT as well as UNIX platforms where multiple servers are accessing the same mail spool. • DSMTP BUG FIX: Added check to stop authent_cache from exceeding MAXC, previously setting authent_cache greater than 1000 caused periodic crashes. • DSMTP FIX: fromip_nolimit now works with "fromip_nolimit a,b" • DSMTP FIX: Fixed bug where absoluted footer file path was munged on NT, e.g. C:\dmail\c:\dmail\footer.txt, so now you really can put footer files in directories other than the dsmtp_path directory. • --------- • DSMTP: Added 'X-AutoResponder: mailer-daemon' header entry as appropriate, e.g. on Bounce messages generated by DSMTP. This hopefully helps to stops autoresponders from responding to bounce messages etc. • DSMTP: Added MAPS RBL support (identical to ORBS) with setting maps_action. • DSMTP: Added support for braindead M$ servers that issue 'ETRN @domain' commands • DSMTP: Fixed bug where "check x" auth queries (e.g. where the email client sends the SMTP AUTH command) were cached-retrieved wrongly. • DList: Fixed error message returned when approv command has wrong password. • DSMTP: Cleaned up a few stray (but harmless) 'free without NULL' instances

	• DPOP: fixed LOGON_BATCH back to LOGON_INTERACTIVE • DSMTP: Fixed problem with self-referential aliases not working if the domain was present, e.g. bob: bob@same_domain,fred@another_domain bob now gets a copy of the message as well as fred. • DSMTP Feature: Added spam_dump command to get filtered spam to be appened to a file. So now if you set, spam_dump /etc/spam.txt the file spam.txt will get every message blocked by the spam_filter (where action is reject) appended to it. • DList Fix: digests loose message attachments - now fixed. • DLIST FIX: footer and footer_html settings. For MIME encoded messages add plain text footer as a mime part, previously they were simply put on end. Also where a message is sent twice in both text and html forms, need to add plain text footer as well as adding html footer to html mime part.
2.8d 23 February 2000 NB: Settings added/changed:signal_core, dtablesize, use_flock,tcp_sendsize, preserve_domain(authent_domain) Commands added/changed:	• Fixed Unintentional DSTMP Change: fixed bug with host_domain setting in versions, 2.7p and 2.8a,b and c, where all host_domain settings starting with the same characters were ignored except for the first one, e.g. host_domain mail.fred.com host_domain mail.bob.com host_domain mail only mail.fred.com would be seen, as they all start with 'mail' and it is the first one. • DPOP FIX: if a message ended in just lf.lf dpop did not check for it and passed it through to email client which upset some clients, notably MS Outlook. So now dpop adds, crlf.crlf to the end of messages ending in just lf • DPOP FIX: added new ini setting, use_flock (default true) which when set to false stops dpop from trying to use flock when lock_id is also set. Enables lock_id to work on systems with a stuffed nfs system. • DSMTP FIX: add_footer setting did not check that footer file finished with a line feed. Which meant that a bad footer file resulted in messages joining together. • Fixed DPOP bug: bin_path setting was ignored if set to something other than the drop_path

	• ------- • DSMTP: added new setting, signal_core (unix only), which when set to true stops dsmtp from intercepting common signals like, access violation. This should be useful in getting a back trace of the dsmtp process by sending it a kill -11. • DSMTP fix: preserve_domain true makes dsmtp pretend that authent_domain true is set so that the incoming address is passed to the user lookup for local domains. Because authent_domain can be left out, dpop sees it as false and therefore it also passes the full user login to the auth module for any domain suffix. This means that adding a virtual domain is as simple as adding a host_domain setting to cover it and usernames in the database become the full email address and the drop file name is the same. It can be a nice way to do vdomains. • DSMTP: fixed DNS timeout adjustment log line as times given wrong way around. • DPOP: new setting, dtablesize, set if you wish dpop (and soon dsmtp as well) to use a dtablesize above our suggested maximum of 1024. • DPOP: new setting, tcp_send_size (default 64k) has been added to enable the setting of the TCPIP output buffer size. Previously it was hardcoded to 8k.
2.7q 2 February 2000 NB: Settings added/changed: host_domain Commands added/changed:	• Fixed Unintentional DSTMP Change: fixed bug with host_domain setting in versions, 2.7p and 2.8a,b and c, where all host_domain settings starting with the same characters were ignored except for the first one, e.g. host_domain mail.fred.com host_domain mail.bob.com host_domain mail only mail.fred.com would be seen, as they all start with 'mail' and it is the first one.
2.7p 26 January 2000 NB: Settings added/changed: Commands added/changed:	• DSMTP AND DPOP: fix from 2.8c put in. Fixes handling of tabs and extra whitespace problems in dmail.conf settings which was put in in 2.7. • DSMTP: fixed ras_domain setting. If ras dial up is turned on and no ras_domain setting was set then dsmtp used to crash. Soltuion was to set, ras_domain to a few spaces. Broken only in 2.7 versions, fixed in 2.7p and 2.8c

2.8c 26 January 2000 NB: Settings added/changed: Commands added/changed:POP DELE command	• DSMTP AND DPOP: fixed handling of tabs and extra whitespace problems in dmail.conf settings which was put in in 2.7. • DSMTP: fixed ras_domain setting. If ras dial up is turned on and no ras_domain setting was set then dsmtp used to crash. Soltuion was to set, ras_domain to a few spaces. Broken only in 2.7 versions, fixed in 2.7p and 2.8c • DPOP Feature: added dpop POP command dele all which responds +OK n messages deleted, rather than having to delete individual messages, e.g. dele 1, dele 2 etc.
2.8b 12 January 2000 NB:Fixes NT permission login bug. Settings added/changed:ban_mailfrom,ban_rctpto,reverse_name_ban, etrn_relay,orbs_action,virtual_user_pre,suspend_domain auth_cache Commands added/changed:tellsmtp filters	• DPOP: temporary fix, increase limit on disabled accounts from 100 to 1000 • DSMTP: Fixed 'tellsmtp showq' hang • Fixed DMSetup: now asks user about forward_from_ip setting instead of defaulting to just 127.0.0.1. It now defaults to 127.0.0.1,1.2.3.* where 1.2.3.* is the first part of the machine's ip address (dmsetup detects the ip address). • Fixed Unintentional DSTMP Change: 2.8a did not allow multiple vdomain lines with the same prefix (needed for vdomain aliases), this restriction has been removed. • Fixed Unintentional DPOP Change: 2.8a and 2.7n changed the NT_User login permission needed to, 'log on as batch job' from 'log on locally' and 'access this computer from the network'. • DSMTP: DSNs for errors occurring before the message is sent no longer say, "destination server said" • ---------- • DSMTP: Added rejection pattern based explicitly on RCPT TO and MAIL FROM by adding settings ban_mailfrom and ban_rctpto. • DSMTP: Added reverse_name_ban to block connecting domains • DSMTP: Added etrn_relay command. Relays any ETRN commands to the given ip numbers. • DSMTP: Added ORBS support via orbs_action command (ban only, currently. Will add vanish, reject and forward user@domain) • DSMTP: Added virtual_user_pre support for sendmail's virtual user tables (_post setting pending) • DSMTP: added setting suspend_domain.

	• DSMTP: Added pattern matching to 'tellsmtp filters' • DSMTP: logs clearly when the auth cache index resets or initialised • DSMTP: dsmtp's auth cache size is now settable via auth_cache • DSMTP: trims trailing spaces off domain names in rcpt and mail lines
2.7o 11 January 2000 NB: Fixes NT permission login bug. Settings added/changed: Commands added/changed:	• DSMTP: Fixed bomb_hash fatal bug • Fixed Unintentional DPOP Change: 2.8a changed the NT_User login permission needed to, 'log on as batch job' from 'log on locally' and 'access this computer from the network'.
2.8a 13 December 1999 NB: Settings added/changed: Commands added/changed:	• DSMTP: Fixed bomb_hash fatal bug • DSMTP: Fixed base64 decode bug • DList Fix: when checking for mailer-daemon and postmaster return addresses it is now a case showinsensitive check. Now also checks for these in the return-path as well as the from header. They are checked on messages to all of the mailing list addresses, -request etc. • DSMTP: added auth_hide command, which takes ip numbers (with * and -) or "recentpop" • DSMTP: if auth_allow is present, DSMTP will advertise AUTH • DList: now logs what loglvl each error message is. • DLIST: now logs host and port that it connects to in order to send mail on info logging level. • DSMTP: Fixed bug in comma separated user forward file lines (found as not working when $user was added to recipient list). • ---------- •

2.7n 29 November 1999 NB:This version has been frozen for converting to a Release 2.7 version. The next beta version will be a 2.8 version. Settings added/changed:show_ehlo, auth_allow Commands added/changed: tellpop rebuild	• DSMTP Bug: Killed a bug in the new fromip code - DSMTP died on freebsd machines at DATA stage. • DSMTP Fix: Added protection against runaway user forwardfile recursion - limits to 5 recursions. Previously dsmtp got into a loop when a user had a .forward file which pointed to themselves. • ---------- • DPOP:Added tellpop command, rebuild user, which rebuilds the user's message index files in their bin file. • DPOP: now writes which bulletin a user is up to in their user_inf file • DPOP: tryied to change dpop NT login so that it does not require user log on access, but found to be impossible. • DPOP: changed logging so that it logs channel fro every log line, also uses 3 special channel numbers, -1 in poll loop, -2 shutdown -3 background • DSMTP: Added MS Exchange/Outlook style AUTH LOGIN DSMTP: Altered the EHLO command to reply with a single multi-line packet, to work with buggy MS clients • DMSetup Fix: on unix platforms now check that /usr/local exists - caused a problem on MACOSX. - changed all of the startup scripts to try this line as well for platforms like BSDI, limit coredumpsize 20000 DSMTP: Dwatch not parsing settings using multiple spaces/tabs as command/param delimiter -made show_ehlo and auth_allow into comma separated multi value settings. DSMTP: fixed show_ehlo setting so that dsmtp looks at more than the first value.
2.7m 19 November 1999 NB: Settings added/changed: drop_old,authent_cache,max_sessions,fromip_nolimit ,alias_file_domain	• Fixed DPOP BUG: stop kill ever using -, when an authent process failed it was possible for dpop to issue the kill -1 command which on some operating systems knocks out all processes!!!. • Fixed DPOP BUG: max_sessions was limited on Solaris boxes because of library limit to file handles. This has been fixed - you should now be able to open 1000 concurrent pop connections. • Fixed DPOP BUG: memory leak in dpop reload. authent_cache was not freed on reload. ie about 370k per reload for default of 1k user cache.

- Fixed DPOP bug: - fix bug in reading reply from external auth etc. replies over 200 chars lost start of reply - now it allways keeps first 100 bytes plus last 100 roughly
  NOTE: increased buffer for replies from external auth to 100.
- Fixed DPOP bug: fix tellpop stats number overflows
- Fixed DSMTP bug: msg_filter setting was getting a '/' added to the end of it which stopped it working on some operating systems.
- DSMTP Fix: fromip_nolimit now looks at *both* the ip number and reverse lookup name. So if you have lookup_names set to true any ip addresses you have listed as exceptions to the fromip setting will work. Also you can specify domain names in the fromip_nolimit setting if you have lookup_names set to true.
- DSMTP Fix: alias_file_domain may now take the form 'alias_file_domain domain filename suffix' where suffix is appended to any alias destinations with only subdomain names, e.g.
  bob: fred@machine1
  becomes a legal setting if the suffix was set to .domain.com
  , i.e. fred@machine1.domain.com.
  NB: dsmtp only adds the suffix if there is no dot in the destination domain.
- Fixed DSMTP Bug: Fixed glitch in RCPT and MAIL code that rejects <"blah"@domain.com>
- Fixed DSMTP queue bug: Improved throughput - it was possible for the queue to get filled up with messages not going to be retried for 1 hour.
- Fixed DSMTP Bug: Fixed old (serious) mem leak associated with RCPT code
- Fixed DSMTP Bug: Fixed new (minor) mem leak associated with short-name crash.
- ----------
- DPOP Fix: make it log name of authent and slave processes on first spawning.
- DPOP Feature: tellpop stats now has another collumn lastcon which is time of last connection for each user (secs since 1970 -:)
- DPOP Feature: tellpop list_current now has another collumn which is sent bytes for each current user.
- DPOP Feature:- if drop_old setting contains a seperator then use path rather than tacking onto

	droppath. So dpop will read drop files from another path as well as reading its ones in the hash directory. This lets you move to another hashing method. NB: this does not mean old bin files will be checked so you have to drop all users first. ● DSMTP: Now takes note of manager_ip commands - this new feature is turned off until dpop can conform with setting changes. ● DSMTP: Added authent process lag-tolerance code. So dsmtp can recover from one off slow repsonses from an authentication process and not loose other lookup responses done at the same time. ● DSMTP: Put the *.dmp files on the same rotation scheme as the *.sta files
2.7k 11 November 1999 NB: This version fixes a number of bugs in earlier 2.7 pre beta release versions. Definitely upgrade versions 2.7a-j to this version. We have only listed bugs fixed from the 2.5 line and differences from 2.5k. NB:This version is essential for users running multiple dpops accessing the same drop files! Settings added/changed: preserve_domain, bind_in, bind_out, quota_fix,dump_stats, add_status, smart_reload, (DList settings forward_from_user, footer_html) Changed default behaviour of host_domain and authent_domain Added support for SMTP AUTH command.	● Fixed DPOP BUG: File locking and lock_id setting. Now dslave processes do dot locking. ● Fixed DPOP BUG: multiple dpops used to allow one user to connect more than once resulting in drop file corruption. Now if lock_id is set then dpop creates a drop_dir.x file while user is logged in(between userpass and free on quit or die of chan) and other dpops lock the user out while that file exists. ● Fixed DPOP BUG: Quotas were not being updated correctly (used line in _inf file) after delete of msgs when they had been previously left on the pop server. So did the following to correct: ❍ compact bins - if nmsgs=0 remove bins reguardles set used to 0 ❍ fixed diskquota not being updated correctly after delete of msgs ❍ added new ini setting quota_fix, set this to true for 1 week so affected users are not denied messages. ● Fixed DPOP bug: fixed key bug on 64 bit machines like OSF. ● Fixed DPOP BUG:if fail during drop user then delete the drop file as msgs all still in bins. ● Fixed DPOP BUG: fixed bug in drop user - which resulted in messed up drop files. ● Fixed DPOP BUG: uses more memory than necessary if you have a large users.idx file and are adding users regularly. (was in all versions). ● DSMTP: Changed default behaviour: of host_domain looked up on external

authentication in 2.7e. DSMTP looks up first host_domain for all virtual domains. 2.7i onwards has the setting, preserve_domain true, which makes dsmtp do its previous behaviour. Admins who think that they need this setting should think about whether they should be using vdomain settings instead of vhost settings.

- DPOP: new config code fixes known peculiar behaviour in 2.5g: user-quota setting needs to be set with two settings in this specific order if setting a default user quota,
  user_quota x
  user_quota true
  where x is the default quota.(fixed in 2.7e)
- Fixed DPOP BUG: non unique uidl's for messages burst in same second - can cause dmailweb/cwmail a few problems when displaying lists of messages.
- Fixed DPOP BUG: memory leak in reload of virtual domains in tellpop reload.
- Fixed DPOP BUG: fixed status line appearing on same line as from after drop user on NT. Was upsetting some email clients.
- Fixed DPOP BUG: fixed problem bursting large messages. When you had no dpop_path setting, the slave process was defaulting to 'dmaildslave', ie with no seperator.(all versions up to this one).
- Fixed DSMTP Bug: DNS problems - dns lookups that timeout for a specific domain cause other messages in the queue to get held there. Added cache of failed dns lookups for short period.
- Fixed DPOP BUG: DPOP now passes all environment variables when it spawns the authentication processes.
- DPOP FIX: max_sessions limit increased for NT. If max_sessions requires it use _setmaxstdio to up file handle limit to what they have asked for (limit is 2048 file fopens)
- ----------
- DPOP FIX: make it selectable which ip addresses on a machine dpop binds to, dsmtp also does this with setting, bind_in. DSMTP also has bind_out setting.
- DPOP FIX: Status header; new ini setting add_status - default true. If set to false then dpop does not add the Status header line.
- DPOP FIX: Status header; make status change to read only on retr command and not on the top

command.

- Fixed DPOP BUG: was sending apop stamp on end of the external authentication lookup - which could confuse some modules. Now only sends it if apop_enable is true.
- DPOP Change: for list with 0 msgs send reply and . in one package to speed it up.
- DSMTP: '$user' can now be used in .fwd files, as per other redirection methods. Instructs dsmtp to still deliver to original recipient as well as redirecting as per a forward carbon copy (forward_cc) setting.
- DSMTP: Added show_ehlo command (previously documented as ehlo_show). So show_ehlo a,b,c means dsmtp adveritises a,b,c as well as ETRN,DSN,HELP in response to the EHLO SMTP command, where a,b and c can be any of the key words, '8bit', 'vrfy' or 'auth'. Switches on the following ESMTP notifications: 8bit, auth and vrfy
- DSMTP: Added support for AUTH commands, use setting, 'auth_allow relay', to allow relaying by authenticated logins.
- DSMTP: log_data setting changed to take 'some', as well as true/false. When set to some, dsmtp logs all tcpip transactions except for middle of data section so that your log is not full of attachment data!
- DSMTP: added setting dump_stats. If set to x then dsmtp appends stats information to a file called, dmxxyy.dmp, in the log_path every x minutes.
- DLIST: Fixed dot stuffing of list messages
- LIST:Fixed logging of failed smtp sends, no longer says, 'finish unsuccessfully' when SMTP connection fails.
- DLIST:Added forward_from_user true setting, if set for a list then email is sent with 'from' header of user and 'reply-to' header of list-request.
- DLIST:Fixed handling of HTML -request commands so it usually works.
- DLIST:Added feature, list setting footer_html d:\footer.html which will be added to HTML messages just before the @lt;/html> line.
- DPOP FIX: if cant start first authent process then die and remove pidf so dwatch does not restart us.
- TELLPOP + DPOP FIX: no longer do they talk

about DPOPXTRA commands, but rather manager commands.

- Fixed DPOP BUG: no longer reports bursting error to users who connect before any mail has ever been sent to them.
- TELLPOP FIX: tellpop shutdown etc now return quickly if manager_ip_address etc wrong, rather than waiting for a timeout.
- DSMTP: Added smart_reload boolean setting. When true, DSMTP and DPOP will try to create a temp.conf in the work directory by merging #included files as/when appropriate, spreading the load out so it occupies at most 20% of dsmtps attention during the process. If it fails for any reason, it will do a normal reload. If you include more than 500 files into dmail.conf then we suggest that you use this setting as reloads could take as much as 3 minutes to complete.
- DList: fixded listname@domain style of naming domains in lists.dat (used to look in drop file without domain on front) NB: all mailing list drop files go in the drop_path dir not vdomain dirs.
- DSMTP FIX: now works with new dlist syntax, listname@domain. Any explicit domain settings will override it, however.
- DList: fixed max_size setting in lists.dat - was being ignored. Default is kbytes and any letters put on end of setting are ignored, e.g. 40k = 40M = 40kbytes.
- DList FIX: Made sendlog not crash it if no parameter
- DPOP FIX: max_sessions limit increased for NT. If max_sessions requires it use _setmaxstdio to up file handle limit to what they have asked for (limit is 2048 file fopens)
- DPOP BUG: now ignores case when checking for "External" for auth method to enable cacheing.
- DPOP FIX: fixed user_quota setting again, so that user_quota x turns on quota system as well as user_quota true.

2.5k 15 Sept. 1999 NB: Version 2.7b will be up soon with new features and recent small bug fixes. Version 2.5k has major bug fixes in it only. Settings added/changed: from_test	• from 2.7b,Fixed DSMTP Bug: multiple aliases for the user were not all being actioned, only the last one was. • from 2.7b,Fixed DSMTP Bug: forward slashes '/' in usernames are relayed correctly and rejected nicely for local addresses (currently they are not permitted in local usernames!) • from 2.7b;DPOP BUG: tellpop drop in 2.5i could merge messages. • from 2.7b; DPOP FIX: Added setting, from_test, defaults to false. If set true then DPOP's from checking is not as fussy about From headers in drop files - you may want to use this when dpop is running with SMTP servers other than DSMTP. (If true checks for From instead of full test of From line.) • Freeze on startup on RH6 bug fixed: from 2.7b,- sendmail stub: no longer freezes on, sendmail -bi as sent by RH6.0 start up script. • from 2.7b,- sendmail stub now responds to mailq with use tellsmtp showq (rather than tellsmtp que).
2.5i 12 Aug 1999 NB: This version is no longer on the ftp site. Please upgrade DSMTP to the one from 2.5k if you are using this version. NWAUTH Version: 2.0g Settings added/changed: allow_dup_drop,vdomain_substitute	• • DSMTP: before forward slash '/' was not allowed in usernames, now it is (as per the relevant RFC). • DSMTP: added setting, allow_dup_drop. If true and two recipient lines point to the same drop file, e.g. an alias or forward point to a drop file (rather than a user) then two messages end up in the drop file. Currently only gets delivered. • DSMTP FIX: added tighter control on the vdomain feature, 'vdomain masquerading' where the host_domain is replaced with the appropriate vdomain in message headers. Replacement no longer done in the From: header (if you want masquerading here then you need to configure the email client properly). Added setting vdomain_substitute (default true) so that you can turn off this feature all together.

2.5h
not released

NB:
NWAUTH Version: 2.0g

Settings added/changed: fake_vrfy

- DSMTP BUG: handle left dangling on hitting fromip_max and possibility for two channels to talk to same file in 2.5h versions, in 2.5g versions the error showed itself in a safer form where DSMTP rejected any connections and filled the log file with messages about failure to open .tmp files in the work path.
- DSMTP BUG: fixed fatal crash with invalid syntax of destination address.
- DPOP BUG: fixed >from bug, in DPOP
- DSMTP: reduced log lines sent when sending data out.
- DSMTP FIX: improved dsmtp outgoing send on windows platforms so that each channel sends for at least 20 milliseconds. So faster channels can send more chunks without interuption.
- DSMTP: made vrfy optional with setting, fake_vrfy true - will say yes to everything.
- DWatch: added sendlog command (for DMAdmin)
- DWatch: added support of dimap, on NT imapd (and imapdsvc on NT)
- DSMTP FIX: adding incoming/outgoing/tcpip channel information to log messages.
- NWAUTH: removed funny search behaviour when an @ in the search (it was put in to handle search *@domain nicely), now use search @domain to achieve same effect.
- DPOP: DSMTP's timezone setting now used by dpop for bulletin Date: field.
- DPOP:fixed bulletin Date field format
- DPOP: improved nt authentication failure messages.
- TELLSMTP: improved tellsmtp sendlog command so stays open until you kill tellsmtp - e.g. ctrl-C
- NWAUTH: nwauth updated to version 2.0g
  - FIXED BUG: part or all of nwauth.txt could get lost during rebuild, added lockfile handling during rebuilds to fix.
  - added Lynden's -fallback_unix option
  - removed funny search behaviour when an @ in the search (it was put in to handle search *@domain nicely), now use search @domain to achieve same effect.
  - made file_config use arg_path when no dmail.conf on machine.
  - made logging always append

	- added encrypt option - added fallback_unix option
2.5g 7 July 1999 NB: Settings added/changed: use_forward_files,gateway,tellsmtp monitors	• New DMSTP setting: added setting use_forward_files true/false. When set false all checking for forward files (.fwd and .forward files) is turned off. • Fixed DPOP: dpop would start but would ignore incoming connections when restarted by dwatch on its death. • Fixed DSMTP: on windows NT dsmtp was putting up a debug box on fatal errors (when it died), which was stopping dwatch from restarting it. • Fixed DPOP bug: death when compacting bin files when bin file could not be opened • Fixed DSMTP bug: self referencing aliases introduced in 2.5f all worked except for when the self referencing destination came last, e.g. bob: test@test,bob now works as an alias. • Fixed DPOP: (small memory leak) made ini file stuff free memory when reading in multiple settings. • Fixed DLIST bug: 2.5d had handle test left in which affected performance - removed again in 2.5f. • Fixed DSMTP: accept nulls in messages, and try to continue as normal. • Fixed DSMTP: EMail clients or servers that send multiple lines without waiting for a response would cause that TCP channel to go into sulk mode. • Fixed DSMTP(affects DList); if lists.dat reads, "list listname" then it used to try to write to drop file, \dmail\in\ listname , i.e. with a space! • Fixed DSMTP: was writing drop files in incorrect directory when multiple external authentication lookups for the same message - e.g. when you had two destinations on different domains for an alias. • Fixed DSMTP: channels receiving log lines do not timeout (although tellsmtp sendlog does timeout). Previously in DMAdmin you might have noticed that it kept losing connection to the DSMTP server every so often. • Fixed DSMTP: (UNIX and default case settings) messages arriving for mixed case usernames

were being put in drop file, named with case as per username in address rather than as per case in passwd file.

- Fixed DPOP: temporary fix for >From lines in message body (messages were being split into two on tellpop drop).

- DSMTP: added tellsmtp command, tellsmtp monitors which shows you all channels receiving log lines (e.g. channels to dmadmin).
- DSMTP: tellsmtp showchans now tells you if a particular channel is receiving log lines (e.g. DMAdmin).
- DSMTP: when dsmtp starts sending log lines out on a channel it starts by telling you what logging level of messages it is sending.
- DSMTP: when a TCPIP channel times out the log file warning message, tells you how long it was open for.
- DPOP: if no stats_path then don't show 0's for kilobytes served etc.
- DLIST: changed error message, 'Invalid destination user bob@domain, SHOULD PROCESS THIS PROPERLY' to read, 'Invalid destination user %s, you might like to REMOVE them from the users.lst file.' Message is given when user address on list cannot be delivered.
- DLIST: changed some info level log messages to debug log level so that on startup from command line you don't see so much rubbish.
- DSMTP feature: all log lines now stay on one line (any carriage returns replaced with spaces).
- DSMTP feature: (gateway setting) can now gateway to domain names (used to be just to ip addresses).
- DSMTP feature: make message id use domain in MAIL FROM: if a valid vdomain instead of host_domain.
- DSMTP feature: postmaster messages for non local delivery bounces appear to come from vdomain.
- DWATCH feature: added a timeout on dwatch's port testing

2.5f
8 June 1999

NB: With this we will be looking to end the 2.4 line at 2.4k

Settings added/changed: forward_from, max_sessions
DList: reply_to_user

Special Mention:
DNAuth update,
Improved moderators subscribing users to mailing lists.

- Fixed DPOP BUG: max_sessions was being limited by false file handles limit (because of check stopping at 256). Now dpop checks you have got enough file handles to give max_sessions requested. NB: Max. sessions can still be limited by the compile time DTABLESIZE variable, contact support if you are having problems.
- Updated DNAuth to 1.0b:
  - - build index did not handle blank fields - implemented token_split
  - - added -version command line option.
  - - fixed bug with rebuild being done twice at startup, done_first
- Fixed tellpop command: made it respond to tellpop pass command
- Fixed DMAdmin bug: temporary file, dmail.in being left open on pressing config button.
- Fixed DMAdmin bug: Large dmail.conf files not being sent saved properly. (tcp_setblock not succeeding).
- Fixed DMAdmin text: forwarding tab did state "last match is applied" now states, 'all matches apply'
- Improved DMAdmin; made it remove its previous '#Adding stuff here' lines on rewriting dmail.conf.
- Changed forward_from setting: made case insensitive so,
  forward_from domainx.com
  covers, DOMAINX.com Domainx.com etc.
- Enhanced DMSetup: if user chooses to limit manager ip addresses, always add 127.0.0.1 on start of list to stop them locking themselves out.
- Added DList feature: setting reply_to_user can now take an address, as well as true/false. If given, posted messages will have any Reply-To: header turned into X-Reply-To:, and the address given is added to a new Reply-To: header.
- Enhanced DList: now wont read in lines from users.lst that don't have an '@' symbol in the address field - this implies they will be removed as on re-construction only those users read in are written out again.
- Fixed DList Bug: users who send subscribe (or unsubscribe or leave) command to listname instead of listname-request, get subscribed, but

	everyone on the list (or the moderator on a moderated list)was also getting the subscribe message they sent. ● Fixed DList bug: subscribing users incorrect names being put in full name field, now takes, subscribe "real name" <address> subscribe <address> "real name" ● Fixed DList bug: when user subscribed themselves and no real name attainable, real name used to take on moderator's real name - now that field is left blank. ● Enhanced DList: when log rotates, puts line at the top like, 28 11:31:16 DList log rotated (60kbytes) 2.5e ● Fixed DSMTP bug introduced in 2.5e: caching always returns 'out of sync' - which meant that it always did a real lookup. ● Fixed DSMTP bug introduced in 2.5d: any aliases to file (specifically DList aliases) were corrupted, so no messages to mailing lists could get through. ● Fixed DSMTP weirdness: no longer inserts too many X-Rcpt-To entries when talking to robots
2.5e 20 May 1999 Due to the following bugs this version has been REMOVED from the beta directory -Mailing lists not working in 2.5e -DSMTP caching not working in 2.5e Settings added/changed:	● Fixed DPOP bug: file handles being inherited so dpop.log not rotating when running nwauth or dslave. ● improved DSMTP: made dsmtp check that external authentications respond with the correct user that was given in the lookup. DSMTP also checks that cache lookups return the correct username. ● - added DSMTP feature: made messages from postmaster where possible, have a from address of, postmaster@virtual_domain rather than postmaster@first_host_domain. ● - fixed DSMTP bug in 2.5d: has a bug with External Authentication modules returning fwd="", e.g. when you modify a user using DMAdmin. If the fwd="" field does not have a destination in it then DSMTP was trying to forward the message to the blank address. As a result DSMTP would return the error,"no valid rcpt fields". This has been fixed in 2.5f (not yet built), so contact support-dmail@netwinsite.com if this bug is affecting you.

2.5d
14 May 1999

NB: The 2.5 line includes New Features the 2.4 line only has bug fixes.

Settings added/changed:
forward_from, dotstuff_robot

- added DSMTP feature: Sleepy external authentication processes now timeout
- fixed DSMTP bug: Fixed NT handle leak for robots introduced in pre-release of 2.5d (so only a couple of people have this version with the handle leak in it - and they have probably already upgraded).
- fixed DSMTP bug: Stopped DSMTP from locking up when authents sulk
- fixed DSMTP bug: NT robot reading code to work with non-responding robots
- changed DSMTP behaviour: Made DSMTP slaughter innocent robots on shutdown (after asking nicely)
- fixed DSMTP bug: now recognises that external auth responses not starting with '-' may not be valid
- fixed DSMTP bug: recognises setting lines with only \t (tab) delimiters
- changed DSMTP behaviour: dotstuffing now defaults to true on NT, false otherwise... and it works.
- added DSMTP feature: NULL environment vars aren't set
- fixed DWATCH bug: hard loop when dwatch restart of a server is 'disabled'.
- fixed bug DLIST and DWatch: stopped them writing all socket stuff to a file raw.dat - you may like to search for a file called raw.dat on your system (probably in the dwatch or dlist or work_path directories) - if it is there then it may be very big.
- Fixed DPOP bug: On authent_method nt_user, cache of last lookup, not doing lookup when last check failed.
- Fixed tellpop bug: no longer creates empty dpop_.sum files on tellpop stats command, and instead writes the output to a file, dpop.sum
- Updated dmsetup:asks dwatch to stop on all platforms even if no pid for it in dwatch dir (as old versions did not write pid).
- changed DSMTP behaviour: now *any* user lookup that gets back a fwd="blah" field will generate a 250 User OK response, no matter whether the forward rule works or not.
- Updated dmsetup: asks dwatch to stop on all platforms even if no pid for it in dwatch dir (as old versions did not write pid).

- Made DSMTP hide domain names (and replace with vdomain names) for robots and direct-to-file deliveries, as it does on normal write to drop file.
- Fixed DList bug: Now only the list moderator can unsubscribe users when access_leave set to moderator. Members cannot even unsubscribe themselves.
- Fixed DList bug: Moderator can now unsubscribe a list member when access_leave set to member.
- Fixed Dlist bug: dlist did not realise Rcpt To: had been accepted when an SMTP server responded with 251 instead of 250.
-
- Changed DSMTP back: DSMTP bounces now get all the normal relaying rules applied to them. For 2.5c they did not have any relaying restrictions, now you must set something like, forward_from_ip 127.0.0.1 so that DSMTP can send bounces.
- - DSMTP now gives a 4xx error if the authent processes fail - authent processes can now return -DEAD reason (for example if they cannot access their database), if they do then DSMTP gives the 4xx error so that the sender knows to try again later.
- - DWATCH altered startup messages so not so worrying :-)
- - DSMTP no_dotforward command to stop DSMTP looking for ~/.forward files, this makes it look for .fwd files instead.
- - DSMTP set environment variables on NT when spawning a robot as have always done on UNIX, e.g. MAILFROM, RCPTTO and MSGSIZE
- - DSMTP write X-Rcpt-To: header when feeding to robots,for 2.5d
- - DSMTP feature (for 2.5d) where you can have multiple addresses in a fwd="" ext. auth field, and use $user, as an entry so that the original recipient also gets the message
- Added DNAuth external authentication module beta - reads users from DNews's users.dat file
- Fixed drespond crashing bug: when subject or from in command line empty.
- Fixed DSMTP bug: SMTP connection was hanging when the following combination

occurred, 1. multiple mail redirection (e.g. two forward rules for the same alias) 2. one of mail redirections was to a robot 3. using external authentication. This bug could often be fixed by putting the forward rule for a robot before any other mail redirection rules - yes it is black magic :-)

- Fixed DPOP bug: messages with no body were being skipped on look through bin file (in with message size of -1), i.e. were not counted or retrievable by email client.
- Fixed obscure DPOP bug: blank line in message body of last message in drop file deleted (Netscape mail deletes any trailing or leading blank lines in message body).
- Fixed DSMTP bug: forward from still looking at HELO line and not Mail From line, now checks, mail from, then helo, and accepts all <> (i.e. bounces) for relay.
- known DList bug: Bug in dlist digest operation, when a user removes themselves from the digest they don't get the digest messages that existed between the last one they were sent and their new message.
- Fixed DSMTP bug: DSMTP dying on MAIL FROM:<@domainx,@domainy:user@domainz> lines
- Changed DSMTP behaviour: DSMTP did dotstuff robot mails... now it doesn't by default on UNIX, but still does on NT
- Added DSMTP setting: DSMTP dotstuff_robot true (default is false) to make new version compatible with current robots that expect dotstuffed lines.
- fixed DSMTP possible bug: DSMTP was not finding user home directories properly, see forward files.
- Fixed DPOP bug: wrong path was given for drop_path ~/inbox type drop path setting. Should become, home_dir/inbox where home_dir might be, /home/username.

2.5c
(24 Mar 1999)

NB: The 2.5 line includes New Features the 2.4 line only has bug fixes.

Settings added/changed:
apop_enable (apop not yet finished)

- Fixed spawn problems for Windows 95/98 (also in 2.4k)
- -fixed DSMTP bug: forward from still looking at HELO line and not Mail From line, now checks, mail from, then helo, (As does 2.4j) and accepts all <> (i.e. bounces) for relay (NB: this last bit is an error fixed in 2.5d
- Made DPOP/dwatch (dsmtp already did) delete dpop.exit file on startup and also after it was found while running. Putting a program.exit file in the dwatch_path directory will make that server shutdown as soon as it can in a nice way.
- Modified tellsmtp and tellpop to suggest that the DSMTP and DPOP respectively are not running when they cannot open a TCPIP port to them.
- Modified DMSetup
  - waits at end of install
  - suggests not to worry about files not found.
  - automatically starts dwatch at end of install and upgrade on Win 95/98
  - made it able to shutdown dwatch when not running as a service on Win 95/98
  - now automatically sets up nwauth on Win 95/98
  - fixed bug, dlist.exit being put in dlist path.
- Added first stage of APOP, on external authentication.
- Added setting, apop_enable true, which results in DPOP sending
  'user username digest ipaddress apop_stamp'
  to the external authentication routine. More details once it is completed.
- Fixed DList bug: archiving count not being recorded - so dir command now works.
- Fixed DSMTP bug with quotes in forward rules and aliases. Aliases should always have quotes around whole robot alias - can't have multiple word arguments until version 2.5c. Forwards must have quotes around them if want multiple words before 2.5c. Now can have quotes around robot and multiple word arguments (quotes within quotes) in either. Aliases still have to have quotes around whole robot destination, but it is optional to have quotes around robot destination in forwards.

2.5b (not released) NB: The 2.5 line includes New Features the 2.4 line only has bug fixes. NB: DPOP2.5b does not work - delete it if you have it on your machine. Settings added/changed: (DList: skip_postmaster_check,skip_mailer_check)	• Fixed multiple forward rules/ext. auth bug from 2.4j • Added dlist list.dat settings, skip_postmaster_check and skip_mailer_check, to stop DList from ignoring messages from POSTMASTER and MAILER-DAEMON (all in capitals forms only). • Changed Dlist default max_per_user setting to 200 from 50. • Fixed Dlist bug: lockup if TCP connection on port 7111 was opened but nothing sent. - seemed like DMAdmin did it occasionally. • Fixed DList bug, introduced with new default individual behaviour, if a list delivery fails on one address, did not move onto next address in list. • Fixed DList bug: non-members can post in period before they have sent back their join cookie. • Fixed DList moderator passwords - password protected moderated lists were not working :-) • Fixed DPOP bug: can't handle more than three >'s in the body of a message, the an extra one would be added by DPOP. • DSMTP dying bug fixed - special (robot or direct to file) recipients causing to crash
2.5 (not released) NB: The 2.5 line includes New Features the 2.4 line only has bug fixes. Settings added/changed: forward_cc, forward, block_domain	• Altered DPOP message error message to email clients from "-ERR external authenticator timed out" to "-ERR password check failed (authentication (external) timed out)". • forward and forward_cc can now take multiple entries and comma delimited lists. • Added DSMTP setting: block_domain <users.domain> <nasty.domain> stops all incoming mail for users.domain if it's from nasty.domain • When using RAS dial up, DSMTP now sends an ETRN command for each of the vdomain domains, and not just the host_domains. • DSMTP now adds Received: from xxx message header lines where xxx is the correct domain or virtual domain, rather than just as set in the first host_domain setting.

2.4k
27 April 1999

NB: This version has Bug Fixes to 2.4j only (new features are in 2.5 line)

Settings added/changed:
forward_from, dotstuff_robot

Special Mention:
Dlist, DWatch and DMSetup in this version are based on the versions from 2.5c

- Made DSMTP hide domain names (and replace with vdomain names) for robots and direct-to-file deliveries, as it does on normal write to drop file.
- Updated DList to the one from 2.5c then added the following three bug fixes,

  1. Fixed DList bug: Now only the list moderator can unsubscribe users when access_leave set to moderator. Members cannot even unsubscribe themselves.

  2. Fixed DList bug: Moderator can now unsubscribe a list member when access_leave set to member.

  3. Fixed Dlist bug: dlist did not realise Rcpt To: had been accepted when an SMTP server responded with 251 instead of 250.
- Updated dwatch from 2.5c
- Updated dmsetup from 2.5c
- Fixed drespond crashing bug: when subject or from in command line empty.
- Fixed DSMTP bug: SMTP connection was hanging when the following combination occurred, 1. multiple mail redirection (e.g. two forward rules for the same alias) 2. one of mail redirections was to a robot 3. using external authentication. This bug could often be fixed by putting the forward rule for a robot before any other mail redirection rules - yes it is black magic :-)
- Fixed DPOP bug: wrong path was given for drop_path ~/inbox type drop path setting. Should become, home_dir/inbox where home_dir might be, /home/username.
- Fixed DSMTP spawn problems for Windows 95/98 (also in 2.5c)
- Fixed DSMTP bug: DSMTP dying on MAIL FROM:<@domainx,@domainy:user@domainz> lines
- Changed DSMTP behaviour: DSMTP did dotstuff robot mails... now it doesn't by default on UNIX, but still does on NT
- Added DSMTP setting: DSMTP dotstuff_robot true (default is false) to make new version compatible with current robots that expect dotstuffed lines.

2.4j 12 Mar 99 NB: This version has Bug Fixes to 2.4h only (new features are in 2.5 line) Settings added/changed: forward_from, show_8bitMIME, no_rcvd_ip Special Mention: 8 Bit MIME - changed default behaviour	• changed default behaviour!!! - DSMTP no longer advertises its acceptance of 8 bit MIME messages in its welcome banner. You can make it go back to doing by setting, show_8bitmime true in dmail.conf • Fixed DSMTP crashing bug: on multiple forward rules for the same user when using external authentication. • Fixed DSMTP crashing bug: due to bad fprintf • Fixed forward_from setting so that it uses domain given in SMTP envelope, rather than domain given in HELO line. • Fixed file lock problem, spawned processes (nwauth) taking file handles on NT. Evident when start dsmtp from dos window and then CTRL-C to quit, could not restart dsmtp as dmail.lck file being held open by nwauth sub-processes. • Fixed DList bug: where if one list delivery fails (e.g. destination user not found) dlist did not continue on with next address on the list. • Added setting, no_rcvd_ip IPaddress, hides that ip address if it was going to be in the received line that DSMTP adds to the message - for keeping secret your internal IP addresses- which are probably made up and not registered. NOTE will do this for BOTH incoming and outgoing messages. • made DSMTP happy with all dlist_ settings - it was complaining in dsmtp.log that it did not know some of the new ones. • Fixed DSMTP bug, relay_to and gateway settings case sensitive. • Fixed DSMTP bug: lowercase_usernames true stopping fromip_nolimit from working as it should.
2.4i 23 Feb 99 Settings added/changed: domain setting in lists.dat user_quota	• Fixed DPOP and DSMTP new setting bug: user_quotas are now turned on when user_quota 1000 (1Mbyte) as well as user_quota true. • Fixed DList: list specific domains. When you put domain domainx.com in the lists.dat file, then that list is domain specific. DList had to be made to look for the drop file, domainx.comlistname (no separator), instead of just listname, to match the alias that DSMTP creates. This behaviour can be turned off with list.dat setting, list_global true. • Fixed DPOP bug: spawn of authentication

	processes on UNIX platforms was not separating out command line arguments - e.g. nwauth -log would never get started. • Fixed DPOP bug: if spawned authentication process died, DPOP died on printing an error message. Together with above bug, when trying to get nwauth to create a log file ('authent_process /dmail/nwauth -log'), DPOP would die repeatedly.
2.4h (16 Feb 1999) (also known as 2.4g2) Bug Fix Version. Settings added/changed: all forward_* settings, forward_from	• DMAdmin now logs to file, dmadmin.log, if you check the debug check box on the DWatch tab. • Changed DSMTP Relaying behaviour, to suit new manual :-) default relay behaviour is now false, i.e. do not allow any relaying other than as specified, if there are ANY forward_* type relay settings. Previously forward_from_ip was the only setting that turned relaying restrictions on, i.e. all other settings were only exceptions to the forward_from_ip rules. • Fixed DMSetup bug - created garbage work|bulletins path. • Fixed forward_from Bug, it now works off MAIL FROM: line in the SMTP envelope as advertised.
2.4g (12 Feb 1999 - released on Linux and NT only.) Settings added/changed: no_xdpop_header, tarpit_start, tarpit_except, dlist_rotate_log, (lists.dat settings: max_per_hour and footer).	• Fixed DPOP BUG in quota system, used not being reset, when all messages deleted. • Fixed forward & forward_cc commands with chrooting for robots - they were forgetting the domain for the robot to chroot to. • Fixed Sendmail stub, no longer adds domain, if users gives fred@domain. • Added DSMTP tarpit anti-spam feature: tarpitting is where the server begins to respond slower and slower to excessive posting from a single ip/session. Added settings: tarpit_start : commences tarpitting from the numberth RCPT line per session. tarpit_except : allows exceptions to the rule. (note the setting is currently NOT tarpit_exclude as this used to state) • Fixed dlist_rotate_log dmail.conf setting - now takes notice of it :-) whereas it was hard coded to an approximate 60k) - default and minimum value is now 60 K. • Fixed DList bug, hardcoded max. messages per user ever, now it is per hour.

	• Added DLIST lists.dat setting, max_per_user x Sets the max. number of messages allowed to be posted to all lists on the server per user per hour. Note that the count is per user for posts to all lists, whereas setting is per list, so the count is global but whether it applies to a list is list specific (the default is 50). • Changed user_quota to ALSO take numerical value. So user_quota true - turns on quota system, no limit if not specified in username_inf in drop file directory. user_quota 30 - turns on quota system, limit of 30 Kbytes imposed on user's disk usage. user_quota false/0 - turns quota system off. • Added DList lists.dat list setting, footer xxxx where xxxx is the full path to the footer file to be added onto the end of all list messages. • Corrected DPOP's response to no_xdpop_header true setting.
Current Release Version 2.4f (29 Jan. 1999)	• Fixed DSMTP bug: crashing somewhere during a tellsmtp reload if a config setting was too long (alias_file_domain too long). • Fixed DPOP Bug: introduced in 2.4e, hashing was being worked out and logged but not used ! :-(• Fixed unreported DPOP bug and changed setting definition: when lowercase_username set to true, DPOP would lower case the full drop path as well as drop file name. This would probably only have caused it to be a different drop file directory on UNIX platforms and if a mixed case drop file path was being used. Now when lowercase_username is true, DPOP always writes the user name into its cache in lower case. This has resulted in a slight change in setting definition, see lowercase_username. • Note: This version does not support DList domain specific lists, 2.4j and 2.5c both definitely do.

2.4e (26 Jan. 1999) (now removed from ftp site!)	• Added settings: max_rcvd (see below) • Fixed DSMTP Bug: where username matches virtual domain prefix. Previously if the virtual domain prefix was 'sales_' and you had a valid user called 'sales_bob', then if dm_set.htm#drop_prefix was set to false, when a message arrived for the domain where that prefix applied, DSMTP would remove the sales_ from bob's username assuming it was a virtual domain prefix, before writing the drop file. Note: if a user has the virtual domain prefix in their username, you do have to add their name to the password file as for example, sales_sales_bob! • Fixed BUG: postmaster alias was case sensitive. DSMTP automatically, adds this alias as per the SMTP RFC to its internal alias list for each domain. Now it is happy with Postmaster@domain, or postmaster@domain or even PoStMaStEr@domain :-) • Fixed DSMTP bug: system timezone being stamped as three digits, e.g. -600 now stamped as -0600 • Changed DSMTP: to use lowercase_username setting, was just a DPOP setting. • Changed: DPOP and DSMTP so they no longer hash (hash_spool 1 or 2) with upper case letters, e.g. for hash_spool 2 DPOP used to hash, mail/F/R/FRED, now it hashes to mail/f/r/FRED. • Fixed BUG: where DSMTP and DList unconditionally lower cased dropfile names. • Fixed BUG: when noautohost was set to true, DSMTP would not open a connection to itself, ever :(• Added Feature: DSMTP now detects if a message has too many Received: lines, as another method to stop loops. For example this should put a stop to a message going around indefinitely if you have two DMSPT servers gatewaying a message back and forth to each other. • Added the max_rcvd DSMTP setting to set the max. number of received lines allowed in a message, (the default is 15).

2.4d (15 Jan. 1999)	• Added config setting: no_autohost <boolean> This setting switches off DSMTP's automatic adding of host_domain entries for MX lookups which eventually refer back to itself. Default is false. See the DSMTP Settings list • Re-instated config setting: forward_from <wildcard> This relaying setting which was removed in about version 2.1k, has been re-instated in a slightly stronger form. It now checks the from line of the message envelope for the specified domain. See the DSMTP Settings list • Bug fixed: where the last list entry in lists.dat wouldn't get aliased, so was not recognised by DSMTP as a valid list. • Bug fixed: where the list alias files wouldn't get hashed, e.g. DSMTP placed mail for the list cars (hash_spool set to 2) in /mail/cars, rather than /mail/c/a/cars where DList was looking. • Bug fixed: NT spawns inherited socket and file handles which they should not have, e.g. robots and authent processes used to hold TCP connections open until they terminated. • Bug fixed: NT spawns were not killable by DSMTP. • Bug fixed: *nix (Unix'ish) problem with dead processes not being de-zombied - i.e. properly removed from the processes list. • Bug fixed: temp license key expiry date bug, could display dates like 31-FEB, so now shows 1-month, e.g. 1-Mar • Bug fixed: vdomain_separator stuck on '_', now works in dpop - previously it was always set to '_' for DPOP but not for DSMTP. Affects password list usernames for virtual domains, e.g. you can now have usernames in your password list for a virtual domain like, dom1#bob, if you set vdomain_separator to '#', previously could only have had dom1_bob. • Bug fixed: ETRN gateway bug, gateway messages were queued using the IPaddress instead of the domain name when gatewayed. Result: when DSMTP received for example ETRN bob.com, it would not realise that it had any messages for bob.com as they were all queued to go to the corresponding gateway IP address. • Bug fixed: vrfy bug, DSMTP was returning, user@domain@domain to the SMTP command

	"vrfy". Problem was related to vrfy section of code only, everywhere else DSMTP was treating usernames correctly. • Bug fixed: problem with host_domain/vdomain ambiguity, if admin entered host_domain setting and vdomain setting for the same domain, vdomain settings were ignored which resulted in virtual domain mail being delivered locally, unless you were using external authentication. • Bug fixed: log messages appearing about mail bombs when fallback_address did not exist and was pointing at account on the same domain. • Bug fixed: authent_domain true, for users on the main domain DPOP authenticated with user@domain_last, where domain_last was your last host_domain setting in dmail.conf. Now DPOP uses the domain specified by the dpop_host setting, if it exists, otherwise it uses the first host_domain setting. • Chris finally nailed weird memory bug :-)
. . .	• These will be filled in soon :-)
2.2p (10 Aug 98)	• Added tellsmtp command: profile. This displays some DSMTP program statistics, cpu usage etc. • Added dmail.conf setting: lock_id nnn, this can be used when running multiple copies of dsmtp/dpop over NFS volumes, this makes dmail use an internal locking mechanism that will work over NFS. Each DMail should have lock_id nnn in its dmail.conf file, where nnn is a different integer in each case.
	For update information on older versions, see updates1.htm

Products | Downloads | Prices | Support | Company

External Authentication Module, NWAuth

This fully useable example external authentication module comes in all distribution sets. The source is provided on all platforms and for Windows and most Unix based platforms it is pre-compiled, as nwauth.exe or nwauth.

If you do compile it, it must be compiled with 32 bit compiler. Example compile line,
gcc -o nwauth -g -w -Dunix nwauth.o -lc
Note: if you get crypt errors you may need to add, -lcrypt to the end of the line (e.g. on RedHat6 and above).

An example authentication program for use with DPOP and DMAIL. This program is spawned as a sub process by both DPOP and DSMTP (i.e. two processes run simultaneously). DPOP and DSMTP send commands to authent stdin via a pipe and receives output from authent stdout via a pipe. Authent simply reads commands with gets and writes replies with printf BUT fflush must be used after each printf.

Commands:

For the defintion of the External Authentication Protocol (to which nwauth adheres) see the External Authentication section.

Notes:

- NWAuth is case insensitive for the usernames. So to avoid using up users on your license, with for example three bobs, being bob, BOB and Bob, you should probably set lowercase_username true. This also means that you cannot end up with mixed case drop file and bin files on UNIX platforms - so well worth doing.
- Version 2 and above of nwauth adds users to an intermediate user file, nwauth.add, when this reaches 3kBytes, the entries in this file are added to nwauth.txt. This makes user addition much faster on large user databases.
- **NWAuth and NFS drives** Important informtation for those using NFS drives

See the Performance Page for information about NWAuth's efficiency with 100,000 users.

Below is the source distributed with version 2.5f:

```
/*
todo:
        When change or set command, add to '.add' file
        Make database file a command line option.



Sample NetWin Authentication routine, high speed simple text file based
authentication

Command line options:
```

```
        nwauth  -set username[@domain] password INFO
        nwauth  -mod username[@domain] INFO
        nwauth  -check username password
        nwauth  -lookup username
        nwauth  -del username
        nwauth  -search string

Normal processing options when used as external authent module for dsmtp/dpop:
        exit
        +OK shutting down authent process (RHP nope it just goes away but thats ok)
        set user password fwd="fred" groups="adults" name="Mr \"Cool\" Smith"
        check username[@domain] password fromIPaddress
        +OK username@domain drop_file_path uid
        lookup username[@domain]
        +OK username@domain drop_file_path uid fwd="a@b" groups="adults"
        set user password aaa="bbb" ccc="ddd"
        +OK    user added/modified
        delete user
        +OK    user removed
        search string
        +DATA    user1 info
        +DATA    user2 info
        +DATA    user3 info
        +DATA    user4 info
        +OK

Normal processing commands
        set user password [INFO]
        set user (NULL) [INFO]   (change info but leave password unchanged)
        check username[@domain] password fromIPaddress
        lookup username[@domain]
        delete user
        search string

Works by using a file nwauth.txt in this format
        user:password:info

  Changes History:

        TRW added: -log option instead of -debug, only log to file,
                                        dsmtp_path\nwauth.txt
        TRW fixed bug:

        Regan Added :
                -Label EXTERNAL_LOG, which disables nwauth log routines and enables
other
                log routines to be called ie routines in log.c etc...
                - if NOAUTHMAIN then myprintf are replaced with buf_print
          buf_clear and buf_get are companion routines
        Regan Changed :
                -Now prints +OK after recieving quit command.
                -ncpy is now static as conflicted when included with other .c files
that also included
                 ncpy.
2.0b
```

```
        29/4/99 for 2.5d TRW changed:
                - \n on end of +OK
                -
        5/5/99  still for 2.5d, TRW changed;
                - fflush after +ok in response to exit, for most other
                responses it is in the end of the do_command function.
2.0c
        11/5/99 for 2.5d TRW changed;
                - made db_check and search check that build_index and check_add
                        had been done at least once (for wadduser as is CGI).
                - added -version command
2.0d
        17/5/99 for 2.5f TRW changed
                - added -size command line option
                - made log file get logged to dmail dir (or as set by path
                command).
                - added more error logging in search.

*/
/* #define NOAUTHMAIN */
/* #define EXTERNAL_LOG*/

#define VERSION "version 2.0d"
#include
#include
#include
#include
#include
#include
#include
#include
#include
#include
#ifndef WIN32
#include
char *strlwr(char *s);
#endif

#ifdef WIN32
#       include
char *crypt(char *key, char *salt);
#       define getpid _getpid
#else
char *crypt(const char *key, const char *salt);
#endif
int db_check(char *user, char *pass, char *info);
char *auth_info(char *argv[], int i, int argc);
int lib_date(char *fname);
char *value_encode(char *s);
#include "nwauth.h"
int auth_rebuild(void);
void do_vers(void);

char *file_config(char *fname);
void check_add(void);
int free_index(void);
```

```
int has_changed(void);
int do_command(char *s);
int build_index(void);
int str_hash(char *s, int max);
char * db_pass(char *user);
void log_file(char *s);
void auth_fixup(char *user, char *pass, char *info);
static char *ncpy(char *dst, char *src, int len);

void zfreeall(void);
void zstrfree(char *s);
char *zstrdup(char *s);

char * zstrstrnc(char *s1, char *s2) ;
int zstrncmpnc(char *s1, char *s2, int n) ;
int zstrcmpnc(char *s1, char *s2);

#ifdef EXTERNAL_LOG
#       include "new_log.h"
#else
        void imsg(char * arg_list, ...);
        void emsg(char * arg_list, ...);
#    define dmsg imsg
#endif

#ifdef NOAUTHMAIN
#       define myprintf buf_print
#else
#   define myprintf printf
#endif

#define BFSZ 1000
#ifndef FALSE
#       define FALSE 0
#       define TRUE (!FALSE)
#endif
static int isdebug;
static int islog;
static int isend;
static int pid;
static int done_check_add;
static int done_build_index;
#define DFLT_REBUILD_SIZE 3000
static int rebuild_size;

static char *buffer = NULL;

int f_size(FILE *f);
int auth_init(void)
{
        return build_index();
}

int buf_print(char *arg_list, ...)
{
```

```
        va_list arg_ptr;
    char *format, *p;
        char text[512];
    size_t size = 0, new_size = 0;

        va_start(arg_ptr, arg_list);
        format = arg_list;
        vsprintf(text, format, arg_ptr);

        if (buffer) size = strlen(buffer);
        new_size = size + strlen(text);

        buffer = realloc(buffer, new_size + 2);
        p = buffer + size;

        strcpy(p, text);
        return size;
}
int buf_clear(void)
{
        if (buffer) {free(buffer); buffer = NULL;}
        return 1;
}
char *buf_get(void)
{
        if (buffer) return buffer;
        else return "";
}
static char arg_path[BFSZ];
#ifndef NOAUTHMAIN
int main(int argc, char *argv[])
{
        char bf[BFSZ];
        int t,i;
        t = 1;
        srand(time(NULL));

        for (i=1;argc>i;i++) {
                if (strcmp(argv[i],"-debug")==0)
                        isdebug = TRUE;
                else if (strcmp(argv[i],"-log")==0)
                        islog = TRUE;
                else if (strcmp(argv[i],"-version")==0){
                        do_vers();
                        return 0;
                }
        }
        /*if (argc>t) {
                if (strcmp(argv[t],"-debug")==0) {
                        t++;
                        isdebug = TRUE;
                }
                if (strcmp(argv[t],"-log")==0) {
                        t++;
                        islog = TRUE;
```

```
                }
        }*/
        pid=getpid();
        if (isdebug)
                dmsg("\n**DNAuth Started in debug mode, %s, pid=%d
exe/log_path={%s}\n",VERSION,pid,file_config(""));
        if (islog)
                dmsg("\n**DNAuth Started in logging mode, %s, pid=%d
exe/log_path={%s}\n",VERSION,pid,file_config(""));

        if (!build_index()) return 1;
        check_add(); /* Check for new/changed entries */

again:
        if (argc>t) {
                if (strcmp(argv[t],"-add")==0)  {if (t+3>=argc)
auth_set(argv[t+1],argv[t+2],auth_info(argv,t+3,argc));}
                else if (strcmp(argv[t],"-set")==0) {if (t+2>=argc) goto badp;
auth_set(argv[t+1],argv[t+2],auth_info(argv,t+3,argc));}
                else if (strcmp(argv[t],"-mod")==0) {if (t+1>=argc) goto badp;
auth_set(argv[t+1],"(NULL)",auth_info(argv,t+3,argc));}
                else if (strcmp(argv[t],"-del")==0) {if (t+1>=argc) goto badp;
auth_del(argv[t+1],FALSE);}
                else if (strcmp(argv[t],"-check")==0) {if (t+2>=argc) goto badp;
auth_check(argv[t+1],argv[t+2]);}
                else if (strcmp(argv[t],"-lookup")==0) {if (t+1>=argc) goto badp;
auth_lookup(argv[t+1]);}
                else if (strcmp(argv[t],"-search")==0) {if (t+1>=argc) goto badp;
auth_search(argv[t+1]);}
                else if (strcmp(argv[t],"-path")==0) {if (t+1>=argc) goto badp;
strcpy(arg_path,argv[t+1]); t+=2; goto again;}
                else if (strcmp(argv[t],"-size")==0) {if (t+1>=argc) goto badp;
rebuild_size=atoi(argv[t+1]); t+=2; goto again;}
                else if (strcmp(argv[t],"-log")==0) {t++;  goto again;}
                else if (strcmp(argv[t],"-debug")==0) {t++; goto again;}
                else if (strcmp(argv[t],"-help")==0) {
                        myprintf("Usage:\n");
                        myprintf("\tnwauth -set user password var1=value ...\n");
                        myprintf("\tnwauth -mod user var1=value ...\n");
                        myprintf("\tnwauth -del user\n");
                        myprintf("\tnwauth -check user password\n");
                        myprintf("\tnwauth -lookup user\n");
                        myprintf("\tnwauth -search string\n");
                        myprintf("\tnwauth -size x -... (sets max size of
nwauth.add)\n");
                        myprintf("\tnwauth -log -... (turns on logging to
nwauth.log)\n");
                        myprintf("\tGive no switches to run in slave mode\n");
                }
                return 0;
badp:           myprintf("Wrong number of paramters\n");
                        fflush(stdout);
                return 0;
        }
```

```
        for (;!isend;) {
                if (fgets(bf,BFSZ-1,stdin)==NULL) break;
                /* Check if file changes (once a second at most) */
                if (has_changed()) build_index();
                check_add(); /* Check for new/changed entries */
                do_command(bf);
        }
        myprintf("+OK\n");
        fflush(stdout);
        return 0;
}
#endif
void do_vers(void)
{
        myprintf("+OK nwauth %s\n",VERSION);
        fflush(stdout);
}
char *auth_info(char *argv[], int i, int argc)
{
        static char info[BFSZ];
        strcpy(info,"");
        for (;inext) {
                        /*TRW,2.5f; if (strlen(u->user)==0) continue;   */
                        if (strlen(u->user)==0) { dmsg("debug: empty slot in database
(deleted/changed user).\n"); continue;}
                        if (zstrstrnc(u->user,match)!=NULL) goto showit;
                        if (zstrstrnc(u->info,match)!=NULL) goto showit;
                        if (strcmp(match,"*")==0) goto showit;
                        if (strcmp(u->user,match)==0) goto showit2;
                        continue;
showit2:
                        dmsg("ERROR: strcmp matched %s %s\n",u->user,u->info);
                        continue;
showit:
                        if (strlen(domain)>0)
                                if (zstrstrnc(u->user,domain)==NULL) continue;
                        printf("+DATA %s %s\n",u->user,u->info);
                        found++;
                }
        }
        printf("+OK Search Complete %d items found\n", found);
        return found;
}
int auth_search_old(char *match)
{
        FILE *f;
        char bf[BFSZ];
        char orig[BFSZ];
        char user[BFSZ], domain[BFSZ];
        char xinfo[BFSZ];
        char *s,*out;
        int found = 0;
```

```
        f = fopen(FILE_NWAUTH,"r");
        if (f==NULL) {
                emsg("-ERR Unable to open %s, %s\n",FILE_NWAUTH,strerror(errno));
                return FALSE;
        }

        strlwr(match);

        s = strstr(match,"@");
        if (s != NULL) {
                strcpy(domain, s);
                *s = '\0';
        } else strcpy(domain, match);

        dmsg("auth_search: matching {%.200s:%.200s} (always lwrcase)\n",match,
domain);

        for (;!feof(f);) {
                if (fgets(bf,BFSZ-1,f)==NULL) break;
                strcpy(orig,bf);
                strlwr(bf);
                if (strstr(bf,match) != NULL && strstr(bf,domain) != NULL) {
                        s = strtok(orig,":\n"); if (s==NULL) continue;
                        strcpy(user,s);
                        s = strtok(NULL,":\n"); if (s==NULL) continue;
                        s = strtok(NULL,"\n");
                        if (s==NULL) s = "";

                        if (!db_check(user,NULL,xinfo)) continue;

                        out = value_encode(user);
                        printf("+DATA %s %s\n",user,s);
                        found++;
                }
        }
        fclose(f);
        f = fopen(FILE_ADD,"r");
        if (f!=NULL) for (;!feof(f);) {
                if (fgets(bf,BFSZ-1,f)==NULL) break;
                strcpy(orig,bf);
                strlwr(bf);
                if (strstr(bf,match) != NULL && strstr(bf,domain) != NULL) {
                        s = strtok(orig,":\n"); if (s==NULL) continue;
                        strcpy(user,s);
                        s = strtok(NULL,":\n"); if (s==NULL) continue;
                        if (strcmp(s,"(delete)")==0) continue;
                        s = strtok(NULL,"\n");
                        if (s==NULL) s = "";

                        if (!db_check(user,NULL,xinfo)) continue;

                        out = value_encode(user);
                        printf("+DATA %s %s\n",user,s);
                        found++;
                }
```

```
        }
        if (f!=NULL) fclose(f);

        printf("+OK Search Complete %d items found\n", found);
        return found;
}
int lib_date(char *fname)
{
        struct stat st;
        FILE *f;

        f = fopen(fname,"r");
        if (f==NULL) return 1;
        fstat(fileno(f),&st);
        fclose(f);
        return (int) st.st_mtime;
}
int has_changed(void)
{
        static int last;
        int t;
        static time_t last_time;
        if (time(NULL)==last_time) return FALSE;
        last_time = time(NULL);
        t = lib_date(FILE_NWAUTH);
        if (last==0) last = t;
        if (t!=last) {
                last = t;
                return TRUE;
        }
        return FALSE;
}

User *auth_find(char *user);
int free_index(void)
{
        int i;
        User *u,*unext;
        for (i=0; inext;
                        zstrfree(u->user);
                        zstrfree(u->pass);
                        zstrfree(u->info);
                        free(u);
                }
                users[i] = NULL;
        }
        zfreeall();
        return TRUE;
}
#ifdef WIN32
#include
#endif
char *file_dmail(void)
{
#ifdef WIN32
```

```
        static char sysdir[BFSZ];
        static char binpath[BFSZ];
        GetSystemDirectory(sysdir,BFSZ);
        sprintf(binpath,"%s\\dmail.conf",sysdir);
        return binpath;
#else
        return "/etc/dmail.conf";
#endif
}
char *file_config(char *fname)
{
        char *s;
        char var[BFSZ];
        char val[BFSZ];
        FILE *f;
        static char bf[BFSZ];
        static char path[BFSZ];
        if (strlen(path)==0) {
                f = fopen(file_dmail(),"r");
                if (f==NULL) return fname;
                for (;!feof(f);) {
                        if (fgets(bf,BFSZ-1,f)==NULL) break;
                        s = strtok(bf," \t\r\n"); if (s==NULL) continue;
                        strcpy(var,s);
                        strlwr(var);
                        s = strtok(NULL," \t\r\n"); if (s==NULL) continue;
                        strcpy(val,s);
                        if (strcmp(var,"dsmtp_path")==0) {
                                strcpy(path,val);
                        }
                }
                fclose(f);
        }
        if (strlen(arg_path)>0) sprintf(bf,"%s/%s",arg_path,fname);
        else sprintf(bf,"%s/%s",path,fname);
        return bf;
}
int build_index(void)
{
        FILE *f;
        int h;
        char *s;
        char bf[BFSZ];
        int nlines=0;
        User *u;

        done_build_index=TRUE;
        free_index();
        f = fopen(FILE_NWAUTH,"r");
        if (f==NULL) {
                imsg("Unable to open %s, %s\n",FILE_NWAUTH,strerror(errno));
                return TRUE;
        }
        for (;!feof(f);) {
                if (fgets(bf,BFSZ-1,f)==NULL) break;
```

```
                u = malloc(sizeof(User));
                s = strtok(bf,":\n \t"); if (s==NULL) continue;
                strlwr(s); /* Store usernames in lowercase */
                u->user = zstrdup(s);
                s = strtok(NULL,":\n \t"); if (s==NULL) continue;
                u->pass = zstrdup(s);
                s = strtok(NULL,"\n"); if (s==NULL) s = "";
                u->info = zstrdup(s);
                h = str_hash(u->user,MAX_HASH);
                u->next = users[h];
                users[h] = u;
                nlines++;
        }
        fclose(f);
        imsg("Read %d lines from nwauth.txt\n",nlines);
        return TRUE;
}
void auth_fixup(char *user, char *pass, char *info)
{
        User *u;
        int h;
        int dodel = FALSE;

        if (strcmp(pass,"(NULL)")==0) pass = db_pass(user);
        u = auth_find(user);
        if (strcmp(pass,"(DELETE)")==0) dodel = TRUE;
        if (u!=NULL) {
                if (dodel) {
                        dmsg("debug: blanking user field {%s}\n",u->user);
                        strcpy(u->user,"");
                        return;
                }
                dmsg("debug: modified user {%s} passsword or details\n",u->user);
                zstrfree(u->pass);
                zstrfree(u->info);
                u->pass = zstrdup(pass);
                u->info = zstrdup(info);
                return;
        }
        if (dodel) return;
        dmsg("debug: adding slot for {%s}\n",user);
        u = malloc(sizeof(User));
        u->user = zstrdup(user);
        u->pass = zstrdup(pass);
        u->info = zstrdup(info);
        h = str_hash(u->user,MAX_HASH);
        u->next = users[h];
        users[h] = u;
}
int f_size(FILE *f)
{
        struct stat st;
        fstat(fileno(f),&st);
        return st.st_size;
}
```

```
void check_add(void)
{
        FILE *f;
        static int last_size;
        int n=0;
        char bf[BFSZ];
        char user[BFSZ];
        char pass[BFSZ];
        char info[BFSZ];
        char *s;

        done_check_add=TRUE;
        f = fopen(FILE_ADD,"r"); if (f==NULL) return;
        if (f_size(f)==last_size) goto doreturn;
        last_size = f_size(f);


        /* Reread the file, and apply all changes */
        for (;!feof(f);) {
                if (fgets(bf,BFSZ-1,f)==NULL) break;
                s = strtok(bf,":\n \t"); if (s==NULL) continue;
                strlwr(s); /* Store usernames in lowercase */
                ncpy(user,s,100);
                s = strtok(NULL,":\n \t"); if (s==NULL) continue;
                ncpy(pass,s,100);
                s = strtok(NULL,"\n"); if (s==NULL) s = "";
                ncpy(info,s,BFSZ-1);
                n++;
                auth_fixup(user,pass,info);
        }

doreturn:
        fclose(f);
}


char *str_encode(char *s)
{
        char salt[BFSZ];
        salt[0] = abs(rand() % 25) + 'a';
        salt[1] = abs(rand() % 25) + 'a';
        return crypt(s,salt);
}

char *value_encode(char *s)
{
        static char bf[2000];
        int i;
        char *out=bf;
        for (i=0; i<2000 && *s!=0;i++,s++) {
                if (*s=='<') { strcpy(out,"<"); out += strlen(out); }
                else if (*s=='>') { strcpy(out,">"); out += strlen(out); }
                else if (*s=='&') { strcpy(out,"&"); out += strlen(out); }
                else *out++ = *s;
        }
```

```
        *out++ = 0;
        return bf;
}
int auth_set(char *user, char *pass, char *info)
{
        /* Append to file */
        FILE *f;
        int rebuild = FALSE;

        check_add(); /* Check for new/changed entries */
        imsg("auth_set(%s,xxxx)\n",user);

        f = fopen(FILE_NWAUTH,"r");
        if (f!=NULL) fclose(f);
        else {
                f = fopen(FILE_NWAUTH,"w");
                if (f!=NULL) fclose(f);
        }
        f = fopen(FILE_ADD,"a");
        if (f==NULL) {
                emsg("-ERR Could not append to %s %s\n",FILE_NWAUTH,strerror(errno));
                return FALSE;
        }
        if (strcmp(pass,"(NULL)")==0) pass = db_pass(user);
        else pass = str_encode(pass);

        fprintf(f,"%s:%s:%s\n",user,pass,info);
        auth_fixup(user,pass,info);
        myprintf("+OK %s added to database\n",user);
        if (rebuild_size<1) rebuild_size=DFLT_REBUILD_SIZE;
        if (f_size(f)>rebuild_size) rebuild = TRUE;

        fclose(f);
        if (rebuild) {
                remove(FILE_ADD);
                auth_rebuild();
        }
        return TRUE;
}

int auth_set_old(char *user, char *pass, char *info)
{
        /* Append to file */
        FILE *f;

        imsg("auth_set(%s,%s)\n",user,pass);
        if (strcmp(pass,"(NULL)")==0) pass = db_pass(user);
        else pass = str_encode(pass);
        auth_del(user, TRUE);
        f = fopen(FILE_NWAUTH,"a");
        if (f==NULL) {
                emsg("-ERR Could not append to %s %s\n",FILE_NWAUTH,strerror(errno));
                return FALSE;
        }
        fprintf(f,"%s:%s:%s\n",user,pass,info);
```

```
        myprintf("+OK %s added to database\n",user);
        fclose(f);
        return TRUE;
}

int auth_exists(char *user)/*used by wadduser to check if a user already
exists*/
{
        char bf[1000];
        if (db_check(user,NULL,bf)) return TRUE;
        return FALSE;
}

int db_check(char *user, char *pass, char *info)
{
        /* Use index to look up user */
        char userlc[BFSZ];
        int h;
        User *u;

        if (!done_build_index) build_index();
        if (!done_check_add) check_add();


        if (pass==NULL)
                imsg("db_check(%s,'NULL')\n",user,pass);
        else
                imsg("db_check(%s,xxxx)\n",user);

        strcpy(userlc,user);
        strlwr(userlc);
        h = str_hash(userlc,MAX_HASH);

        for (u=users[h];u!=NULL;u=u->next) {
                if (strcmp(u->user,userlc)==0) {
                        strncpy(info,u->info,BFSZ-1);
                        info[BFSZ-1] = 0;
                        if (pass==NULL) { return TRUE;}
                        if (strcmp(u->pass,crypt(pass,u->pass))==0) return TRUE;
                        imsg("Password wrong (%s) (%s)!=(%s)\n",user,u->pass,pass);
                        return FALSE;
                }
        }
        imsg("User not found (%s)\n",user);
        return FALSE;
}
User *auth_find(char *user)
{
        /* Use index to look up user */
        char userlc[BFSZ];
        int h;
        User *u;

        strcpy(userlc,user);
        strlwr(userlc);
```

```
        h = str_hash(userlc,MAX_HASH);

        for (u=users[h];u!=NULL;u=u->next) {
                if (strcmp(u->user,userlc)==0) {
                        return u;
                }
        }
        return NULL;
}
char * db_pass(char *user)
{
        /* Use index to look up user */
        char userlc[BFSZ];
        int h;
        User *u;

        strcpy(userlc,user);
        strlwr(userlc);
        h = str_hash(userlc,MAX_HASH);

        for (u=users[h];u!=NULL;u=u->next) {
                if (strcmp(u->user,userlc)==0) {
                        return u->pass;
                }
        }
        return "none";
}

int auth_del(char *user, int silent)
{
        /* Append to file */
        FILE *f;

        imsg("auth_del(%s,%d)\n",user,silent);

        f = fopen(FILE_ADD,"a");
        if (f==NULL) {
                emsg("-ERR Could not append to %s %s\n",FILE_NWAUTH,strerror(errno));
                return FALSE;
        }
        fprintf(f,"%s:(DELETE):(DELETE)\n",user);
        auth_fixup(user,"(DELETE)","(DELETE)");
        if (!silent) myprintf("+OK Deleted user successfully\n");
        fclose(f);
        return TRUE;
}
int auth_rebuild(void)
{
        /* write new file, rename. */
        FILE *fout;
        char fname[BFSZ];
        int ndel = 0;
        int i;
        int n=0;
        User *u;
```

```
        strcpy(fname,FILE_NWAUTH);
        fout = fopen(FILE_TMP,"w");
        if (fout==NULL) {
                emsg("Unable to open %s, %s\n",FILE_TMP,strerror(errno));
                return FALSE;
        }
        for (i=0; inext) {
                        if (strlen(u->user)>0)  {
                          if (fprintf(fout,"%s:%s:%s\n",u->user,u->pass,u->info)<=0)
goto failed;
                          n++;
                        }
                }
        }
        fclose(fout);
        if (remove(fname)!=0) emsg("-ERR remove of %s failed
%s\n",fname,strerror(errno));
        if (rename(FILE_TMP,fname)) emsg("-ERR rename of %s failed
%s\n",fname,strerror(errno));
        return TRUE;
failed:
        emsg("-ERR Error writing new file %s\n",strerror(errno));
        fclose(fout);
        return FALSE;
}


int str_hash(char *s, int max)
{
        int total=0;
        int r;
        int i;
        for (i=0;*s!=0;s++,i++) {
                total += ((*s) * 8)*i;
        }
        r = total % max;
        if (r<0) return -r;
        return r;
}
#ifndef EXTERNAL_LOG
void emsg(char * arg_list, ...)
{
        va_list arg_ptr;
    char *format;
        char output[2000];
    int len;

    va_start(arg_ptr, arg_list);
    format = arg_list;
    len = vsprintf(output, format, arg_ptr);
        myprintf("%s",output);
    log_file(output);
}
```

```
void imsg(char * arg_list, ...)
{
    va_list arg_ptr;
    char *format;
        char output[2000];
    int len;

    va_start(arg_ptr, arg_list);
    format = arg_list;
    len = vsprintf(output, format, arg_ptr);
    if (isdebug) {
                myprintf("%s",output);
                log_file(output);
        }
        else if (islog) {
                log_file(output);
        }

}
void log_file(char *s)
{
        FILE *f;
        time_t  stamp;
        char bf[BFSZ];
        char *p;

        if (!islog && !isdebug)
                f = fopen(file_config("nwauth.log"),"w");/*log last error only*/
        else
                f = fopen(file_config("nwauth.log"),"a");
        if (f==NULL) return;
        /*fprintf(f,"%s",s);
        fclose(f);*/
        time(&stamp);
        strcpy(bf,ctime(&stamp));
        if ((p=strchr(bf,'\n'))!=NULL) *p=0;/*chop off cr*/
        fprintf(f,"%d %s %s",pid,bf,s);
        fclose(f);
}
#endif
#ifdef WIN32
char *crypt(char *key, char *salt)
{
        /* One way encryption of key using salt to perterb */
        /* This function is only intended to make stealing passwords difficult, the
password file should still be protected */
        static char result[BFSZ];
        int i;

        int start = salt[0] + salt[1]*255;
        result[0] = salt[0];
        result[1] = salt[1];
        for (i=0; i<(int) strlen(key); i++) {
                result[i+2] = ((key[i] * (start+1)) % 40) + 'A';
                start += key[i];
```

```
                start *= 3;
                start = start % 32000;
        }
        result[strlen(key)+2] = 0;
        return result;
}
#endif
#ifndef WIN32
char *strlwr(char *s)
{
        char *ss=s;
        for (;*s!=0;s++) if (isupper(*s)) if (*s>0) *s = tolower(*s);
        return ss;
}
#endif


typedef struct MLIST {char *data; struct MLIST *next;} Mlist;
static int zleft,zupto;
static Mlist *mlist,*mcur;
char *zmalloc(int sz)
{
        Mlist *m;
        char *s;
        if (sz>=zleft) {
                m = malloc(sizeof(Mlist));
                m->data = malloc(10000);
                zleft = 10000;
                zupto = 0;
                m->next = NULL;
                if (mcur==NULL) {
                        mlist = mcur = m;
                } else {
                        mcur->next = m;
                        mcur = m;
                }
        }
        s = mcur->data + zupto;
        zupto += sz;
        zleft -= sz;
        return s;
}
char *zstrdup(char *s)
{
        char *n;
        n = zmalloc(strlen(s)+1);
        strcpy(n,s);
        return n;
}
void zstrfree(char *s)
{
        /* Do nothing */
}
void zfreeall(void)
{
```

```
        Mlist *m,*mnext;
        for (m=mlist;m!=NULL;m=mnext) {
                mnext = m->next;
                free(m->data);
                free(m);
        }
        mlist = NULL;
        mcur = NULL;
        zleft = 0;
        zupto = 0;
}

static char *ncpy(char *dst, char *src, int len)
{
        char *xdst = dst;
        int xlen = len;

        for (;(*src !=0) && (len>0); ) {*dst++  = *src++; len--;}
        *dst++ = 0;
        xdst[xlen] = 0;
        return xdst;
}


static const unsigned char charmap[] = {
        0000, 0001, 0002, 0003, 0004, 0005, 0006, 0007,
        0010, 0011, 0012, 0013, 0014, 0015, 0016, 0017,
        0020, 0021, 0022, 0023, 0024, 0025, 0026, 0027,
        0030, 0031, 0032, 0033, 0034, 0035, 0036, 0037,
        0040, 0041, 0042, 0043, 0044, 0045, 0046, 0047,
        0050, 0051, 0052, 0053, 0054, 0055, 0056, 0057,
        0060, 0061, 0062, 0063, 0064, 0065, 0066, 0067,
        0070, 0071, 0072, 0073, 0074, 0075, 0076, 0077,
        0100, 0141, 0142, 0143, 0144, 0145, 0146, 0147,
        0150, 0151, 0152, 0153, 0154, 0155, 0156, 0157,
        0160, 0161, 0162, 0163, 0164, 0165, 0166, 0167,
        0170, 0171, 0172, 0133, 0134, 0135, 0136, 0137,
        0140, 0141, 0142, 0143, 0144, 0145, 0146, 0147,
        0150, 0151, 0152, 0153, 0154, 0155, 0156, 0157,
        0160, 0161, 0162, 0163, 0164, 0165, 0166, 0167,
        0170, 0171, 0172, 0173, 0174, 0175, 0176, 0177,
        0200, 0201, 0202, 0203, 0204, 0205, 0206, 0207,
        0210, 0211, 0212, 0213, 0214, 0215, 0216, 0217,
        0220, 0221, 0222, 0223, 0224, 0225, 0226, 0227,
        0230, 0231, 0232, 0233, 0234, 0235, 0236, 0237,
        0240, 0241, 0242, 0243, 0244, 0245, 0246, 0247,
        0250, 0251, 0252, 0253, 0254, 0255, 0256, 0257,
        0260, 0261, 0262, 0263, 0264, 0265, 0266, 0267,
        0270, 0271, 0272, 0273, 0274, 0275, 0276, 0277,
        0300, 0301, 0302, 0303, 0304, 0305, 0306, 0307,
        0310, 0311, 0312, 0313, 0314, 0315, 0316, 0317,
        0320, 0321, 0322, 0323, 0324, 0325, 0326, 0327,
        0330, 0331, 0332, 0333, 0334, 0335, 0336, 0337,
        0340, 0341, 0342, 0343, 0344, 0345, 0346, 0347,
        0350, 0351, 0352, 0353, 0354, 0355, 0356, 0357,
```

```
        0360, 0361, 0362, 0363, 0364, 0365, 0366, 0367,
        0370, 0371, 0372, 0373, 0374, 0375, 0376, 0377
};

int zstrcmpnc(char *s1, char *s2) {
        const unsigned char *cm = charmap,
             *us1 = (const unsigned char *)s1,
             *us2 = (const unsigned char *)s2;

        while (cm[*us1] == cm[*us2++])
                if (*us1++ == '\0')
                        return (0);
        return (cm[*us1] - cm[*--us2]);
}
int zstrncmpnc(char *s1, char *s2, int n) {
    const unsigned char *cm = charmap,
    *us1 = (const unsigned char *)s1,
        *us2 = (const unsigned char *)s2;

    while (cm[*us1] == cm[*us2++]) {
                if (*us1++ == '\0') return (0);
                n--;
                if (n<=0) return 0;
        }
    return (cm[*us1] - cm[*--us2]);
}

char *zstrstrnc(char *s1, char *s2)
{
        const unsigned char *cm = charmap;
        char *s;
        int n = strlen(s2);


        for (s=s1;*s!=0;s++) {
                if (zstrncmpnc(s,s2,n)==0) return s;
        }
        return NULL;
}
```

Trouble Shooting FAQ

Send DMail Support the right things FIRST time!

---

1. The server is dying (core dump or DrWatson), what should I do . . .

   (or What should I send to DMail Support)

   If one of the DMail servers (DPOP, DSMTP or DList) is dying it will be evident in a number of ways,

   - users cannot connect to the server
   - the dwatch resurrector may be emailing you as the sys admin.
   - you may get 'core' files appearing in the server directories, e.g. /usr/local/dmail (UNIX platforms only)
   - you may get a DrWatson Window popping up

   Note that dwatch is supposed to restart the servers when this happens, by default it only does that 5 times and then gives up watching that server.

   When one of the servers is dying we at DMail support will of course want to know about it because it means that there is a serious bug in our software.

   See the next faq for suggestions on what to send DMail Support...

2. What to Send DMail Support

So here is a list of the things that it might be appropriate to send us. But **please** don't send us a huge email with lots of large attachments, just pick the best information that you have. Sending us your config file and a log or back trace is usually sufficient. Don't forget to tell us your platform and the version you are using.

- Your dmail.conf file - almost always send us this
- The log file (on debug log level if possible, and maybe with log_data true)
- A 'ded' file from the dwatch directory
- A DrWatson log file, e.g. \winnt\drwtsn32.log
- A back trace from a core dump (don't send a core file)
- A trace.log file from the dwatch_path directory (check the date is valid)

And email those to dmail-support@netwinsite.com

Here are some pointers on gathering the above information.

- Set your logging level to debug as soon as you are supicious that something is going wrong.
  To do this edit your dmail.conf file,
  /etc/dmail.conf or \winnt\system32\dmail.conf
  so that the setting log_level looks like this,
  log_level debug
  save the file and then reload both DPOP and DSMTP
- If the bug is something to do with the TCPIP connections on DSMTP you may want to set,
  log_data true
  so that DSMTP logs all TCPIP connections.
  (or 'log_data some' on 2.7 and above versions so that your log does not end up filled with attachment information)
- Send us the relevant log file, e.g. dsmtp.log which you will find in the log_path directory.
- Send us the relevant 'ded' file, e.g. d_1dsmtp.ded from the dwatch_path directory. These are the log files as copied by dwatch when it noticed that the server had died. If a server has crashed a number of times then a couple of these are useful to see if the last thing in the log is the same each time - i.e. they can answer the question, is the server dying on the same thing each time?
- The most useful thing is a back trace. This shows us which function within our program was being run when it died.

  Getting a back trace on NT:
  DrWatson will create a back trace and put it in the file, \winnt\drwtsn32.log, if it notices any program dying. NB: It may ask you if you want a log to be created, which you should make it do.
  DrWatson should be on by default, but you can turn it on in the DMAdmin utility. Click on, Config Dwatch, then select any server and click on the 'Set DrWatson as debugger' button in the pop up window.
  NB: If the drwatson pop up box comes up and waits for you to click OK, then dwatch will

not notice that the server has died and so will **not** restart it until you click on the OK button. So click 'don't popup window when any program dies', then DrWatson will be set so that it automatically creates the log file for you and then closes the dying program. This allows DWatch to restart the server, but you still get the log.

Getting a back trace on UNIX based platforms

Hopefully if one of our programs dies you will find a file named, core (or core.program on some platforms) in the same directory that the program is running in. So look in the following default server path directories,
/usr/local/dmail for dsmtp and dpop
/usr/local/dmail/dlist for dlist
/usr/local/dmail/dwatch for dwatch itself

PLEASE do not send us the core file. Valid information can only be read from it by analysing it on the machine on which it was created.

So to analyse the core file and get the back trace here are a couple of common examples,

Most Boxes (usng dsmtp as an example):
1. cd to the program directory,
cd /usr/local/dmail
2. run gdb with arg1 being the process and arg2 being the name of the core file,
gdb /usr/local/dmail/dsmtp /usr/local/dmail/core
3. now that gdb should be running enter,
bt
this should display a back trace. Send us a cut and paste of the whole gdb session rather than just the back trace bit.
4. enter quit to close gdb

On AIX:
Same as above but use, 'dbx' instead of gdb. You can also use the '-a pid' option to attach to a running process.

On Solaris:
Same as above. Most customers seem to be able to install 'dbx' pretty easily but it is also quite common to have, 'adb' which has a '-c' option that may be the one to use.

On some platforms we had forgotten the compile flag, -g, in versions before 2.8. So the back traces will be useless, e.g. a message like, 'no symbols found' will appear.

Sometimes it is useful to send us a truss of the program as you can run this while the programme is still running, (truss -p pid). Note that this only shows us the system calls (like disk access) that the server makes (as far as we know - someone tell us if we are missing something :-)). So it is not as good as a back trace.

- Send us a trace.log file if the death is in DSMTP. This is a very basic back trace that DSMTP generates when it dies, but it is not nearly as good as a real back trace. DSMTP puts this file in the dwatch_path directory, usually, /usr/local/dmail/dwatch or \dmail\dwatch. Note: you should delete the trace.log file as soon as you have copied somewhere else as DSMTP will not always overwrite it if the death happens again.
- Lastly, check dates on files and look inside them to see that they contain information from the time of the crash

---

3. I tried to upgrade but it did not work . . .

   Normally if something does not upgrade correctly then it means that the installation utility, dmsetup, was not able to stop that part of the server in order to copy over it with the new version.

   So to do the upgrade you must stop that server or program and then manually copy the new executable over the old one - make sure you find the correct old executable to overwrite!

   A few notes that might help:

   1. On NT remember to exit from DMAdmin before you do the upgrade.

   2. On NT if you want to stop the servers and DMAdmin is not responding then you must stop the DWatch service that controls them - you can do this from the control panel, 'services' dialog. If this does not work then you must disable the DWatch service (in the same dialog) and restart the machine, so that when you restart, the servers are not running. At that point dmsetup should be able to upgrade everything without any problems.

---

4. **What does this log (error) message mean?**

   See Deciphering Log Files.

---

5. **I am having a problem with the users ...**

   The following is a list of things to try given you are having problems with the user database. It assumes that you are using nwauth but most of what is says applies to whatever database you are using.

   I can add users (with NetAuth or whatever) but the servers don't recognise them:

   The most likely problem is that the users are being added in the wrong form, i.e. with the wrong prefix or suffix. You should open up your user database (for NWAuth that is nwauth.txt and/or nwauth.add) with a text editor and see the form of username that has been added there and then compare that with the username that dsmtp and dpop are looking up in the appropriate log file - obviously the two have to match.

   To get the log files that you need, edit the dmail.conf setting log_level to read,
   log_level debug
   Then reload the servers (tellsmtp reload and tellpop reload) then send in a message to that user or login to dpop as that user. In the dsmtp.log file (in the log_path directory) you are looking for the

line,
"lookup username ..."
In the dpop.log file you are looking for
"check username ...". It is the username that should match with the username in the user database.

If they don't match there are a number of settings in dmail.conf that effect the prefix and suffix of a username in the user database. In dmail.conf these are either vdomain(the prefix parameter) and vdomain_separator OR authent_domain. The NetAuth manual has a 'Mail Server authentication setup' section with all of the possible settings and what you have to set in each product. Note: if you are using DMAdmin to enter the usernames then you have enter them exactly as you want them to appear in the nwauth.txt file.

If the usernames do match, then either nwauth is returning a bad response or dpop and dsmtp are not running the same nwauth as you are.

So run nwauth from the command line, e.g. assuming you have a user called bob and his password is 'pass' and that your authent_process setting in dmail.conf is c:\dmail\nwauth.exe, then enter,
c:\dmail\nwauth.exe
lookup bob
check bob pass
exit
The response should be '+OK ...' in each case.

You can check that dsmtp is running the authentication process that you have just run by entering,
tellsmtp config authent_process
It should respond with the value of that setting.

6. **I got a bounce (Delivery Status Notification) message from DSMTP ...**

   DSMTP creates a number of messages for sending back to the sender of a message explaining a delivery problem or notifying of delivery success. These are called, DSNs (Deliver Status Notification) messages, and are generally identified by the fact that the sender of the message is the 'postmaster@your_domain'.

   There is a section of the manual on these, Bounces and DSNs.

   Here is the start of a list explaning some common ones ... (ask us to add to this list if you get a DSN that is not listed)

   - **Subject= Possible message loop**
     The error message is generated when dsmtp detects that a message that it is receiving already has received headers stamped on it 15 times, i.e. indicating that the message has been through many servers, and hence probably gone around and around in a loop (as most messages only have 2 or 3 such received lines).

     As the message states this is normally because a message arrives for a given domain, e.g. bob.com, dsmtp looks at its list of host_domains and vdomains, can't see bob.com there so considers it a non-local domain. So then it does a dns lookup on bob.com and sends the message off to the resulting ip address, which is itself. So the same process happens again

and again.

Normally in this situation, the result of the DNS lookup is noted as pointing at itself so it automatically adds that domain to the host_domain list, and delivers the message locally.

However sometimes it does not realise that the DNS lookup points at itself so no auto host_domain addition occurs. The auto host addition can also be turned off with the setting, no_autohost, so check that that is not set to 'true' in your dmail.conf.

Also, if there is a server forwarding or gatewaying (routing) mail for that domain to dsmtp then the DNS lookup will not point at dstmp directly, so again, it cannot automatically add the host_domain line.

The best way to see what is happening is to set,
log_level debug
log_data true
and then do a tellsmtp reload, so that dsmtp logs the full message body which will show you the received lines in the messages. From that you should be able to trace the path of the message.

If in the rare circumstances you wish to allow your messages to have more than 15 received headers then you can set, max_rcvd to a number higher than 15. NB: we do not recommend changing this setting, as in 99.9% of cases it indicates that you have something misconfigured. This setting is fairly new (probably 2.7k onwards) so enter,
tellsmtp config max_rcvd
to see if your version of DMail knows about that setting.

If you cannot find the solution then please email DMail Support with the bounce message that you get, your dmail.conf and also a dsmtp.log file showing the received headers (or a copy of the message showing the headers).

7. **What does this DPOP error message mean ...**

DPOP returns quite a small set of error messages when it does not allow a user to log in. Good email clients pass these messages through to the email client, but note that some don't. So you should always check the dpop.log file to see the real reason that a user cannot connect to the pop server.

NB: a number of the DPOP error messages are simply the messages returned by an external authentication module - this should be obvious in the dpop.log file if it is the case. We're happy to edit the error responses of any of our authentication modules if you wish to make suggestions.

Here is the start of a list explaning some common ones ... (ask us to add to this list if you get a message that is not listed)

- **Cannot open or lock drop file**
  This error message is a general one meaning that DPOP cannot take control of the user's mail box (drop file). This is normally because of somrt sort of file access problem. If this happens every so often then just ignore it. If it happens consistanly or occurs every time that a user tries to retrieve their mail then you need to examine the dpop.log file (found in

the log_path directory) for details. Remember to set log_level debug to get more detailed information in the log file. If the answer is not obvious then you will need to email the log file along with your dmail.conf file to DMail Support.

NB: a common problem in an old version of dpop was that it did not create the directories leading up to the user's drop file. So try sending a message to the user if you are getting this error, because then you can be assured that DSMTP has created the path for that user.

8. 'Database Down' or 'Out of Sync' message with External User Database ...

   An error message in the dsmtp.log file such as,
   ...Out of sync reply from external auth (bob) isn't (fred)...
   or similarly in a bounce message or server connection error message,
   ...User database is down

   Indicates that DSMTP (or DPOP) thinks that your authentication module is responding, e.g. it looked up,bob and thinks that it received back a response for, fred.

   The probable reason for this is that your authentication module was delayed in responding to the lookup request. So that dsmtp sees the response to that request when it goes looking for the response to the following request.

   The time that it waits is set by,
   authent_timeout
   which takes a timeout setting in seconds.

   Also the settings,
   tcp_timeout (DSMTP)
   and
   pop_timeout (DPOP)
   set the timeout on TCPIP connections for DSMTP and DPOP respectively.

   You could check that your authent_timeout setting is long enough to allow any normal slow lookups by your authentication module, e.g. if your database regularly goes offline for a few minutes each day. I suggest that you set it at 30 seconds if you are unsure what to set it at.

   You also need to check that your tcp_timeout and pop_timeout settings are **larger** than your authent_timeout setting. If they are not then the servers can drop the connection before they have finished allowing the authentication module to do the user lookup. That can cause very strange behaviour. We recommend that you leave both tcp_timeout (default 5 mins) and pop_timeout (default 10 minutes) at their default values.

   In version 2.7n (2.7q is the corresponding release version) we did some work on this so that in such a situation DSMTP can 'get back in sync'.

   So if you are using an older version you may want to upgrade to at least version 2.7q.

9. On Windows, DMAdmin just shows lines like 'Lost connection to DSMTP (Select failed () Connection Refused)' ...

   The messages you are seeing in DMAdmin indicate that the admin utility cannot connect to the

dsmtp (and/or dpop) server(s).

It is important to realise that DMAdmin is just an admin utility that connects to the servers when they are running. It may well be that they are running but DMAdmin cannot talk to them for some reason.

You can check if the servers are really running by entering at a command prompt,
telnet localhost 110
quit
to which the DPOP server should respond if it is going.

Similarly entering
telnet localhost 25
quit
checks for DSMTP.

If the servers are not running please send DMail Support your configuration file,
dmail.conf
(typically c:\winnt\system32\dmail.conf or /etc/dmail.conf)
and the following log files,
dsmtp.log
dpop.log
from the log_path directory (specified in dmail.conf).

NB: the most common cause of this is that there is another Mail server running! So **please** do check that you do not have another SMTP or POP server running. When you do the telnet tests above, the DSMTP and DPOP servers will respond with a line including the word, 'DSMTP' and 'DPOP' respectively so that you can tell that they are the server responding. Other servers will respond with similar lines but of course will not mention the names of our products.

If some other servers are running then you need to shut them down and re-run the dmsetup installation utility (which will do an upgrade, 2, this time). You will find dmsetup in the dmtemp directory.

If the servers are running and they are indeed the DMail servers then DMAdmin is probably just having trouble connecting to the servers. So send DMail Support the same files as above, but also add the dwatch.log file (from the log_path directory) and you can click on the 'debug output' check box on the dwatch tab in dmadmin and send us the resulting dmadmin.log file (dmadmin will log to screen the name of the log file it is using).

10. What does the following **System Administrator** message mean ...

Many System Administrator messages are simply copies of bounce or DSN messages, so in addition to any messages listed below check the FAQ above,
I got a bounce message from DSMTP ...

(no system admin messages doc'd at present - email DMail Support if you want one explained)

Products Downloads Prices Support Company

DMail Anti-Spam features

DMail has a host of features for blocking spam.

Most importantly, a mail server should have Relaying Restrictions imposed on it. Note: The default behaviour for DSMTP is not to restrict relaying.

Relaying is where a message comes in for a non-local user, i.e. a user on a domain that is not covered by a vdomain setting, and DSMTP tries to deliver the message to the SMTP server where that user is local. So it has to do a DNS MX lookup on the destination domain, open a connection to that SMTP server and try to give it the user's message. If that SMTP server does not accept the message then DSMTP will bounce the message. Note that if that SMTP server allows relaying from DSMTP then it will also accept that message and try to send it to the end destination.

Given that you have secured your box against unauthorised relaying, most SPAM is to local users. With the settings below you can restrict spamming on your system.

You can place Volume Restrictions on the amount of mail that one user receives, or the number of recipients for the message.

Another way of stopping spam is to add message filtering such that if a message does not apply to a general set of rules then it will be rejected.

Given that you know who is spamming your server you can set up rules to ban connections from certain sources.

Relaying Restrictions

By default DSMTP will allow all relaying, but as soon as you put one relaying rule in dmail.conf, then all relay requests are only carried out if they are specifically covered by a relaying rule.

So to adding relaying restrictions ...

1. Add a **forward_from_ip** setting as it is the best defense against being an Open Relay.

   forward_from_ip: Allows relaying only from a subset of IP numbers. This is matched against the sender's IP address.

   (In versions 2.7q and above DMSetup will add a forward_from_ip setting for your local subnet, so that your default setup will not be an open relay.)
2. Then **add to the forward_from_ip setting** other settings to provide exceptions for relaying.

Use the following commands for relaying exceptions:

(detailed setting descriptions can be found in the Reference Section at the end of the links)

- forward_from: Allow relaying by users sending from a specifed domain.

  NB: Creates a relaying hole for spammers pretending to be from the specified domain which is easy to do.
  If you have this setting for your main domain then Open Relay Databases like ORBS will almost certainly add you to their 'bad servers' list!
- relay_to: Permits exceptions to forward_from_ip for certain domains. This is matched against the domain given in the envelope's RCPT line.
- forward_user: Permits exceptions to forward_from_ip for recent POP users - often called a 'POP before SMTP' system. This allows users to relay mail for the default period of 2 minutes after checking for mail. You can increase this period with the setting, forward_window.

  NB: this handling of this system was improved greatly in version 2.8m. Before 2.8m it works fine for medium sized servers but became inefficient on large servers if the window was set large, e.g. 1-2 days.
- smtp_auth relay: New in 2.8n! Allows the login of users to the SMTP server. Once authenticated the 'trusted' user can be allowed to relay using this setting.

  DSMTP supports the SMTP AUTH command when this setting is added. This allows the user to turn SMTP AUTH on in their email client. SMTP AUTH means that the email client will provide the username and password (same as on POP server) to authenticate on your SMTP server when connecting to send out mail.

  NB: adding this setting will mean that some email clients like Netscape Mail force the users to turn on SMTP AUTH. Generally this is not a problem as Netscape Mail instructs them on how to do it, but it may be confusing to some users.

  If using the forward_user system as well then you should probably set the setting,
  hide_auth recentpop
  so that unnecessary auth lookups are not done.

Volume Restrictions

DMail also allows restrictions to be placed on the volume of messages coming from a particular IP number going through DSMTP per hour.
fromip_max: Restricts the number of messages per hour that DSMTP will accept from an IP number.
fromip_nolimit: Permits exceptions to from_ip_max for certain IP numbers. This applies to the IP number of the sender.
max_rcpts: This sets how many recipients can be sent in a single message before it is rejected.
tarpit_start: This basically starts slowing down responses by DSMTP once x recipients have been reached
for the session. Each new recipient will get a slower and slower response.
tarpit_except: This allows certain ip address to be exempt from tarpitting.

Message Filtering

Message filtering is also available, though it should be used with care. DMail doesn't do logic checks of them so it may be possible to accidently reject everything (!).
msg_filter: Gives a filename containing messages filtering rules. An explantion of those rules is at the other end of the link

---

Banning

You can also straight out ban anyone from a particular IP address from connecting.
ban_ip: Specifies an IP address that DSMTP may not talk to.
ban_mailfrom: This allows you to ban by the "mail from" header and is a pattern match.
So if you wanted to ban bob@aol.com
ban_mailfrom bob@aol.com
or all aol users
ban_mailfrom *@aol.com
ban_rcptto: This allows you to ban by the recipient, it is pattern matching like ban_mailfrom.

Thinking about SPAM ...

If your site gets spammed and say greater than 50% of users get hit, then you really have to ask yourself, 'Where did the spammer get their list of names?' Here are some things that spring to our minds ...

- Through VRFY?:
  We don't think that it is common for spammers to use the SMTP vrfy command to check if usernames exist on your server. There does not seem to be a gain to doing this over simply sending a recipient line which is less work. The spammer can do a mail out and then build up a list from the successful posts on that mail out.
- From Public Posting?:
  Certainly it is true that if your users post to public sites like, News Groups or Mailing Lists then they are likely to get more spam. The same is true for usernames that are common, e.g. we should all have pity on people called, Joe, Bob or Fred :-)
- From your Mailing Lists?:
  If you have not restricted access to the 'who' command on your mailing lists then that might be a way that a spammer has obtained lists of your users
- Your user database has been obtained?:
  The possibility of this happening certainly can not be ignored if you have a large unexplained posting.
- Any suggestions?:
  If you have suggestions for anti-spam features then we will be glad to hear them.

Products Downloads Prices Support Company

SmtpAuth

Overview

SmtpAuth is a small program which is to be used in conjunction with email clients that do not support smtp authentication, but whose smtp server requires it. It runs in the background on the client's machine, accepting incoming unauthenticated smtp requests, and forwarding these to the smtp server, adding in authentication using a username and password supplied during installation. It runs as a service on Windows NT, or as a startup program on Windows 95/98.
Note : SmtpAuth only supports the plain-text authentication method.

Download

SmtpAuth 1.00 - Windows NT/95/98 (58 kbytes)

Installation

1. SmtpAuth consists of a single executable file smtpauth.exe. Download this, and run it by double clicking on it, or typing "smtpauth" at the command prompt. It then prompts your mail server name, followed by your user name and password. Pressing < Enter > for the mail server name will use the default value of "mail".
2. **Once SmtpAuth is running, you must change your email client settings to indicate that your outgoing (smtp) mail server is now "127.0.0.1"**
3. Note that while SmtpAuth is installed, it will automatically restart whenver your computer restarts.
4. To uninstall SmtpAuth, just run SmtpAuth again, and choose the delete option.

Configuration

To change the settings at a later date, just run SmtpAuth again and choose the reconfigure option. By default, SmtpAuth assumes your smtp mail server will be on port 25. If this is not the case, in your windows system directory ("winnt\system32" on nt, or "windows\system" on win95) you will find a file "smtpauth.ini". Change the line that reads "mail_port 25" to whatever port your smtp server is on. SmtpAuth also assumes that it should listen on port 25 on the system that it is running on. This can also be changed by modifying the "port 25" setting in "smtpauth.ini". After modiying anything in "smtpauth.ini", you must restart SmtpAuth. This can be achieved by running SmtpAuth and choosing the restart option.

TITLE

Known Bugs in Current and Recent Versions of DMail

This list is not exhaustive. It is intended to highlight major bugs in the version you are using, to help you to decide if you should upgrade. Most bugs even if very serious only affect systems with a particular setup, so mostly it is quite safe to keep using a version with known bugs.

Known Unexpected behaviour of DMail

These are things that you might expect to work but don't.

- Do not use spaces in file names anywhere in dmail.conf, e.g. do not make your work_path setting,
  work_path c:\program files\dmail
  as certain parts of DMail will break due to our use of space as a delimiter.

Current Beta Version (2.8z2), (implies not yet fixed in any version on site)

- **Too many DSMTP incoming connections - Solaris only:** All versions currently suffer from problem in dsmtp where incoming connections are sometimes processed slowly on solaris only. This leads to many more concurrent incoming connections than normal. Send any reports to DMail Support specifically send the output of, tellsmtp showchans when you can get it. If this problem is occurring on your machine you will see a many channels in state 6 for periods greater than 60 seconds.
- **File Locking Bug on Solaris only:** now does not fail to secure lock on drop files on solaris sparc OS when using lock_id and NFS drive. Used to give 'Permission Denied' error when trying to lock drop file. May need to set, use_flock false, to turn off additional flock calls which will fail on NFS with permission denied error. Fixed in 2.9b, 2.8z2
- **DSMTP and Authent Processes with command line arguments:** Previously only the first authentication process spawned by DSMTP, was spawned with the command line arguments. If you run a system where in your config file you specify command line arguments on the authent_process setting, e.g.,
  authent_process c:\dmail\nwauth.exe -drop_path 2 c:\dmail\in
  then you should work out if those options affect the authent processes's response. If they do then you should upgrade urgently to version 2.8z3. This is fixed in versions 2.8z3 and 2.9b
- **Security Issue - upgrade to at least 2.7r!** See the following page for details,
  DMail Security
- **Security Note:** robots are run as root user if DSMTP cannot ascertain user id to run them as, e.g. when using NWAuth. 2.8l adds setting, robot_defaultuser, see misc. faq
- **Windows 2000 bug:** when no dns_host setting is set in dmail.conf, dsmtp is not able to find system settings for dns servers. On some setups this may cause dsmtp to die (on DSMTP versions before 2.8l). So on windows 2000 you MUST specify your DNS server with dns_host settings.
- **nwauth and NFS:** NFS File locking has not been added to nwauth yet. So you must not store the nwauth.txt and nwauth.add files on an NFS drive IFF NetAuth or some other program is adding users to the database
- **Robots and forward:** robots are not working in Early 2.8 beta versions of dsmtp if set with a 'forward' setting. They do work if set in an alias file or with the fwd="" field. This will be fixed in 2.8l. A domain name is being attached to the end of the forward setting destination.

- **NFS and File Locking:** File locking on NFS mounted drives and shared mail spools still fails for DPOP (DSMTP is fixed). This only causes problems where drop_user or tellpop drop is being used
- **Msg_filter:** DSMTP throws out msg_filter settings after a reload! in 2.8f, to be fixed in 2.8h.
- **Handle Leak:** DPOP known handle leak when authent processes die and have to be respawned - only a problem if something goes very wrong with authent process and it cannot be restarted.
- **MAC/OSX cannot install startup script:** (2.5k onwards)
- **Tabs in dmail.conf:** NT users should avoid using tabs in dmail.conf as DMAdmin does not understand them (all versions)

Current Release Version: (2.7q and 2.7r) (implies fixed in current beta)

- 2.8w and 2.9a, small but possible DList exploit fixed.
- 2.9a, 2.8
- **NFS and File Locking:** File locking on NFS mounted drives and shared mail spools fails occasionally. This has been fixed in version 2.8e. Lock_id should now be used on NT (as well as UNIX) where multiple servers share a mail spool area.
- **ETRN and Microsoft Servers:** Some microsoft servers send 'ETRN @domain' which breaks the relevant RFCs. The fix is in version 2.8e and above of DSMTP.
- **DSMTP Death:** Added check to stop authent_cache from exceeding MAXC, previously setting authent_cache greater than 1000 caused periodic crashes. Fixed 2.8e
- **fromip_nolimit:** fromip_nolimit only works if set on single lines, e.g.
  fromip_nolimit a
  fromip_nolimit b
  (fromip_nolimit a,b does not work). 2.7 and 2.8 versions upto and including 2.7p and 2.8c, fixed in 2.8e
- **Inefficiency in Queue handling:** DSMTP is slow to deliver messages with large numbers of recipients because it uses a single queue file for the message. We are fixing this asap, expect fix to be in beta version 2.8f.

Previous Release Version: (2.5g) (implies fixed in current beta)

- DSMTP bug: On the OSF platform, dsmtp can crash in its mail bomb hashing code. This occurs when high ASCII is received as the sender's address.- Occurs in all versions before 2.8a and 2.7o
- DSMTP bug: if a user has a .forward file that points to themselves, dsmtp ends up in a loop - causing it to die on some platforms - fixed 2.7n
- DPOP BUGS: in drop file locking when running multiple POP servers accessing same drop files. You **must** use 2.7k if you are in this situation.
- DPOP BUG: in quota system. If login and leave mail on pop server, sets 'used' entry in _inf file correctly, but if log in again and delete all mail, used entry stays the same. In all versions fixed in 2.7k
- DPOP known peculiar behaviour in 2.5g: user-quota setting needs to be set with two settings in this specific order if setting a default user quota,
  user_quota x

user_quota true
where x is the default quota.(fixed in 2.7e)

- License keys fail on 64 bit machines like OSF (temporary ones work).
- DPOP: large drop files can get corrupted when dropping them (all versions, fixed in 2.7k)
- DPOP: uses more memory than necessary if you have a large users.idx file and are adding users regularly. (all versions, fixed in 2.7k).
- **fatal bug when invalid destination syntax** (all versions before 2.5h)
- **RH6 (and above) Linux box hangs on startup after installing DMail** (all versions before 2.5k)
- **fromip_max handle leak** (all versions before 2.5i)
- **nwauth loosing user accounts** (all versions before 2.5i)

Recent Beta/Release Versions: (most customers wont have seen many of these beta versions)

- DSMTP: error with password being wrong on SMTP AUTH lookups in 2.8h and possibly earlier versions too. Fixed in 2.8i. Bug manifests itself by allowing AUTH a number of times and then no more.
- NT only bug in 2.8d beta! : Beta version 2.8d should not be run on NT systems where the authentication method is set to use the system user database (authent_method nt_user) as by accident DPOP requires the user permission, 'Log on as a batch job' to allow users to log in. This will be fixed in 2.8e
- Whitespace in the config file is not extracted in early 2.7 and 2.8 versions. So you cannot use tabs within things like forward settings. This is fixed in versions 2.7q and 2.8d.
- If ras dial up is turned on and no ras_domain setting is set then dsmtp will crash. Soltuion is to set, ras_domain to a few spaces. Broken only in 2.7 versions up to 2.7o, Fixed in 2.8c and 2.7p
- 2.8a did not allow multiple vdomain lines with the same prefix (needed for vdomain aliases), this restriction has been removed in 2.8b
- 2.8a and 2.7n changed the NT_User login permission needed to, 'log on as batch job' from 'log on locally' and 'access this computer from the network'. Put back in 2.8b
- DSMTP: Killed a bug in the new fromip code - DSMTP died on freebsd machines at DATA stage. - put in in 2.7m, fixed in 2.7n
- DPOP BUG:2.7 versions up to 2.7m will kill all processes on the machine (kill -1) on some platforms when external authent process dies
- dpop2.7k still has a memory leak of about 300k on reload with authent_cache of 1000- fixed 2.7m
- DSMTP Bug: msg_filter setting was incorrectly getting \ put on endin 2.7 versions, fixed 2.7n.
- DSMTP: minimal channel leak (aggravated by tarpit setting), fixed in 2.7m
- DPOP: max_sessions is limited by file i/o on some systems - only causing a problem on Solaris, all versions, fixed in 2.7m
- DPOP: messages with no body have a blank UIDL (they get one when they get dropped), some clients report those messages as not existing! (in 2.7a-k).
- **Domain Aliases** (all versions before 2.5g)
- **Aliases and Forward rules causing delivery to wrong domain** (all versions up to 2.5g)

- 2.7 only, DLIST: hash_spool 2 does not work for 3 letter list names in 2.7 versions, hashes 'mis' to /var/spool/mail/m/mis instead of /var/spool/mail/m/i/mis. Fixed 2.7h
- 2.7 only, DSMTP: dies if you have a redundant vdomain or host_domain line - i.e. one of each for the same domain. In all 2.7 code, fixed in 2.7i - bug in dmc_removen in dm_conf.c
- 2.7 only, DSMTP: Fatal bug with MX loop notification code - workaround is remove line no_autohost true (in 2.7 only ,fixed in 2.7i)
- 2.7 only, **Check_gid defaulted to 'false' rather than no setting** (2.7a - 2.7d)
- 2.7 only, Default slave_timeout setting was incorrectly set to 10 seconds rather than 100 seconds in versions 2.7a-d, fixed in 2.7e.

---

fatal bug when invalid destination syntax

Consequences: dsmtp dies, and if message makes it into queue files then dsmtp dies every time the message comes up for delivery - yes very bad!

Setup: independent

Found in: all versions before 2.5h, but 2.5k is the current version to replace with.

Details: If a message arrives addressed with a strange syntax DSMTP dies. You have to delete the offending q_ file from the work_path directory to stop dsmtp from dieing.

You should upgrade to version 2.4k

RH6 (and above) Linux box hangs on startup after installing DMail

Consequences:your Red Hat 6 or above box freezes at the point where sendmail is starting.

Setup:independent

Found in:(all versions before 2.5k when installed on Red Hat 6 and above)

Details: The startup script for sendmail changed in Red Hat 6 distributions and unfortunately our 'sendmail stub' does not handle the new way in which it is called.

To get going again, reboot your machine. At the lilo prompt enter,
linux single
so that it is started in single user mode. Then once it has started rename,
/etc/rc.d/init.d/sendmail /etc/rc.d/init.d/sendmail.not
and reboot your machine as normal.

You should either upgrade to version 2.5k or download the sendmail stub for linux_libc6 and replace,
/usr/sbin/sendmail
with it. (NB: on some platforms you also have to replace, /usr/lib/sendmail if it is not a link).

---

fromip_max handle leak

Consequences:handle leak results in dsmtp not being able to open .tmp files in the work area.

Setup:setting fromip_max turned on, and limit reached regularly (possibly only on windows platforms).

Found in:(all versions before 2.5i)

Details:DSMTP stops responding to any TCPIP connections as it has run out of file handles. The log is generally full of messages stating that .tmp and other files in the work area could not be opened for strange reasons.

---

nwauth loosing user accounts

Consequences:

Setup: Using external authentication with the option of nwauth. More than one user being added at a time by any of nwauth wadduser or netauth.

Found in:(all versions before 2.5i)

Details:newer versions of nwauth (with dmail versions 2.4f and above) operate with two user data files, nwauth.txt and nwauth.add, instead of just nwauth.txt. The .add file is used to change entries in the user database without requiring a re-read of the entire nwauth.txt file (which is one line per user). Every time a user is added the .add file is checked to see if it is above the 'rebuild file size' (3kbytes by default). If it is then nwauth re-writes nwauth.txt and deletes nwauth.add.

The problem arises when more than one nwauth (or wadduser) tries to add a user at the same time and the nwauth.txt file is re-written by both of them. There is the potential to loose a section or all of the nwauth.txt file. This is a rare occurrence on most systems.

Domain Aliases

Consequences: DSMTP does lookup of username@alias_for_domain instead of username@main_domain

Setup: External Authentication and authent_domain set to true and multiple host_domain settings (aliases for your main domain).

Found in: All versions up to 2.5g

Details:

With these settings in dmail.conf,
host_domain h_domain.com
host_domain alias_domain.com
authent_domain true

dpop_host h_domain.com

(where the first host_domain setting is defined to be the main domain and the rest aliases for it)

If a message comes in to DSMTP for local delivery to the user,
bob@alias_domain.com
then dsmtp sends the following line to the external authentication,
lookup bob@alias_domain.com

The problem with this is that in nwauth.txt (or nwauth.add) the admin will have only added the user,
bob@h_domain.com

How does dpop handle it?

DPOP has its own dpop_host setting so it ignores all host_domain settings (so does not know, and does not need to know, about the domain aliases) and always looks up tam@h_domain.com. Note: this can cause a problem if the dpop_host setting is set to something different from the first host_domain setting.

Aliases and Forward rules causing delivery to wrong domain

Consequences: DSMTP writes incoming messages to drop file in wrong domains directory.

Setup: External Authentication AND authent_domain set to false AND aliases or forward rules to multiple local destinations on different domains (drop_prefix false might make the consequences worse).

Found in: all versions that handle such aliases up to 2.5g

Details:

If there are two destinations for an alias and each destination requires an external authentication lookup. Then if the destinations are on separate domains DSMTP will probably think that both drop files belong in the second domain's drop file path. If drop_prefix is set to true (the default) then drop files from separate domains will have different prefixes so the error will be obvious when looking in the drop directories. If drop_prefix is false then users with the same username on different domains could get the incorrect mail!

NB: this is an bug unreported by any of our customers.

MAC/OSX installs incorrect startup script (2.5k onwards)

Consequences: On reboot dmail does not start itself up

Setup:N/A

Found in:2.5k and above - hope to have this fixed in version 2.7c

Details: The installation utility installs a startup script for dmail in /etc/rc.d/init.d on UNIX platforms, unfortunately this is not a valid path on MAC/OSX.

See the note on this in this section, Startup Scripts or contact DMail Support.

Check_gid defaulted to 'false' rather than no setting

Consequences: Users get error message about problem with their drop file.

Setup:UNIX platforms only.

Found in:(2.7a - 2.7d)

Details: Fixed default for check_gid to no group. It was 'false' when the setting takes a group name. This means that dpop would not allow access to drop file once it had something in it because gid of drop file will never match 'false'. Hence users will get a message saying that there was a problem with their drop file.

Products	Downloads	Prices	Support	Company

DList - Mailing Lists

NB: You must make DSMTP reload after changing DList lists so that it can create the aliases that it needs for the lists (check for them in the file aliases.dml in the work directory).

DList - Quick Overview

DList is a mailing list server that is part of the DMail package.

General settings for DList are contained in the main configuration file, dmail.conf.

To create a list you simply add a line like,
list listname

setting in the file, lists.dat which you will find in the dlist_path directory. Then make DSMTP reload the configuration file, dmail.conf (DList regularly checks the configuration and lists.dat files for changes, so it does not need to be sent a reload command).

Generally the sysadmin would set up a list, and then users would send an email to the 'listname-request' address to 'subscribe' themselves to the list.

Users in general will only interact with the list by sending emails, either directly to the list to be 'posted', or to the listname-request address if they wish to join the list or send it commands. See:
DList mail commands
When users join the list they are normally sent this list of commands so that they know what the list can do for them.

To modify DList settings you can directly edit dmail.conf and lists.dat with a text editor, use the windows management utility DMAdmin or use the web based tool, NetAuth.

Creating a Mailing List

To create a mailing list on the list server, DList, you need to add a new list setting in the lists.dat file, e.g.
list listname
where listname is the name of the list. You can edit the lists.dat file with a text editor (e.g. notepad or vi), or the windows administration tool DMAdmin will edit the file for you if you use the "Mailing Lists" button.

If you are doing it manually then below the list setting add any other settings that you require for your list, e.g.
title title of my list

Then RESTART the DSMTP server with DMAdmin. You must restart DSMTP after changing DList lists so that it can create the aliases that it needs for the lists.

To try out the list, you should add a user to the list and then post a message to the list. For information on this see,
Adding Users to a List
DList Email Commands

NB: Mailing lists on Virtual Domains...

If you are wanting to add the mailing list to a specific domain, e.g. a virtual domain, then you need to specify that domain in lists.dat, so that DSMTP can create the correct aliases for your mailing list.

There are two ways to do this.

1. Old way: add a domain setting to your list, e.g.

```
list juggling
   title Mailing List
```

domain vdomain1.com

2. New way: create the list with a full list name, e.g.

list juggling@vdomain1.com
title Mailing List

(NB: you **must** use dmail version 2.7q or above, for option 2 to work!)

The second method is better because it means that all mailing list directoryies will be created with unique names. This allows you to reuse mailing list names on different domains.

NB: no matter what domain a mailing list is on, you will find its drop file in the main directory's drop_path (not in the vdomain drop_path), as set by the dmail.conf drop_path setting.

Settings - lists.dat

Lists.dat is the file where you create all the lists on your DList server and where you enter individual settings for each list. (General DList settings, i.e. not list specific, are entered as for DPOP and DSMTP in the dmail.conf file, see Settings - dmail.conf)

As with dmail.conf all settings are one per line, and you can exclude a line by starting it with the '#' character. You do not need to reload the DList server after making changes to dmail.conf (as you do with DSMTP and DPOP), as DList automatically checks the dmail.conf and the lists.dat files before each check for list messages.

Note: after adding a new list to lists.dat you MUST make DSMTP do the reload command. This is because DSMTP has to create aliases for each of the lists on your list server.

Below is a list of all of the settings available for each list. All settings for a list are entered on the lines following the
list listname
line that declares a list, before the next list starts with its 'list listname' declaration line. See example lists.dat file

All settings take just ONE value except where stated otherwise in the description.

Note: In the table below you will see that the 'access' settings can generally take one of the following values. It is important to think about what these settings mean - NOT all of them apply to every access setting! See moderated lists

- member - refers to list members and in general the list moderator as well
- anyone - no restriction
- moderator - only the list moderator can do it
- *domain - person trying to do it must have the email address ending in 'domain'

Setting	Default	Example/options	Description

access_join	anyone	anyone,*netwinsite.com,	Controls who can join the mailing list
access_leave	anyone	in version 2.5d (2.4k) and above: members: (can unsubscribe themselves, moderator can unsubscribe anyone) moderator: (only moderator can unsubscribe - members cannot unsub themselves)	Controls who can unsubscribe from the mailing list, by default anyone can unsubscribe anyone else.
access_post	members	moderator	Controls who can post messages to the mailing list
access_who	members	anyone	Controls who can retrieve the list of current members
archive	false	true	If set, DList will record all incoming messages in an 'archive' sub directory, off the list's directory.
domain	(none>	domain mydomain.com	Specifies the domain that this list should exist on where you do not want it to be on your first host_domain. NB: To allow listname re-use on different domains see the note, Mailing lists on Virtual Domains
footer (version 2.4h and above)	(none)	footer c:\dmail\dlist\listname\footer.txt	The full path to a file that you want added onto the end of all messages as a footer. In version 2.8e and above this is only added onto all TEXT messages, HTML version also added see below.

footer_html (version 2.8e and above)	(none)	footer_html c:\dmail\dlist\listname\footer_html.txt	The full path to a file that you want added onto the end of all HTML messages as a footer.
join_cookie	false	true	If set, when users join the list they will be asked to respond with a specific cookie (number) to prove they are real humans, this setting prevents people from subscribing other people or worse other lists to an existing list. Note: a cookie will not be sent if the subscriber is a moderator or if access_join for the list is set to moderator or password.
list	(none)	dnews-discussion	The name of the list, this cannot contain spaces and must be the first setting for each new list in lists.dat
max_size	40	100	Limits the maximum size of an item that can go through the mailing list in kbytes. NB: this setting applies to messages to the -request address as well as the posting address.

max_per_user (2.4g and above)	200 (changed from 50 in vers. 2.5d)	1000	Sets the max number of messages allowed to be posted to all lists on the server per user per hour. Note that the count is per user for posts to ALL lists, whereas the setting is per list. So the count is global but whether it applies to a list is list specific (the default is 200).
moderator	(none)	fred@netwinsite.com	A list of one or more moderator email addresses, a moderator often has extra access rights, like the ability to subscribe other people etc. Separate multiple entries with spaces or tabs (or commas in version 2.5d and above)
reply_to_user	false	true (also in version 2.5f and above, user@domain)	If set, the reply-to header in each message will be pointed to the original poster, rather than the mailing list, this is recommended for large mailing lists. In versions 2.5f and above in place of true you can specify an address as the reply address for ALL messages posted to the list. If given, posted messages will have any Reply-To:

			header turned into X-Reply-To:, and the address given is added to a new Reply-To: header.
status_interval	7	1	Period in days between automatic status reports being sent to the moderator.
skip_mailer_check	false	TRUE	If TRUE then DList will not ignore messages from users called, MAILER-DAEMON (all in capitals). These are normally bouced messages and so would not normally be wanted as posts to the list.
skip_postmaster_check	FALSE	TRUE	If TRUE then DList will not ignore messages from users called, POSTMASTER (all in capitals). These are normally bouced messages and so would not normally be wanted as posts to the list.
subject_prefix	(none)	Juggle:	This string will be added to the front of every subject line of messages from this list, this makes it easy for people to sort list messages out from other messages.
title	(none)	N.Z. Juggling	A title for the list, shown in headers and lists output.

An example lists.dat file with entries for two lists, talk and juggling:

list	talk	
	archive	true
	title	The list for talkers.
	subject_prefix	[list: talk]
	access_join	Anyone
	access_post	Moderator
	access_who	Anyone
	access_get	Anyone
	moderator	talk.master@macro.com
	max_size	40
	footer	c:\dmail\dlist\talk\footer.txt
list	juggle	
	title	The list for jugglers.
	subject_prefix	[Juggle]
	access_join	Anyone
	access_post	Anyone
	access_who	Anyone
	access_get	Anyone
	moderator	juggling.master@macro.com
	max_size	40
	footer	c:\dmail\dlist\juggle\footer.txt

Welcome Messages

DList comes with an example welcome message. It is stored in a file called join.tpl in a template format.

You can edit this template to the look that you require, and you can copy it to each list's directory (off the main DList directory) so that individual lists can have their own welcome message.

The template can use the following template variables, which will be replaced by DList with the appropriate information. A template variable is parenthesised with double percent sgins, e.g. %%variable%% so that DList can recognise it as a variable.

%%list-name%%	BigList	The name of the list.
%%list%%	biglist@domain.com	The address that list members send messages to post on the list.
%%list-request%%	biglist-request@domain.com	The address that people send list request messages to, e.g. in order to subscribe.

%%h_user%%	bob	The username of the 'person' who has joined the list.

Adding users to a list

Usually users would add themselves to a list, by sending a message to the list request address, e.g. listname-request@domain with the word subscribe in the message body.

Dlist will then add them to the users.lst file for that list. Users.lst for each list is stored in that list's directory (named after the list) off the dlist directory (probably \dmail\dlist\listname\users.lst or /usr/local/dmail/dlist/listname/users.lst)

To add a number of users you have two options:

1. Add yourself as a moderator for the list and send the listname-request address a message with mutliple subscribe lines in the body.

   So as a moderator you send the following email:

   To: listname-request@domain
   From: your_moderator_address

   subscribe bob@domain1.com
   subscribe judy@domain1.com
   subscribe george@domain99.com

   to join up the email addresses,
   bob@domain1.com
   judy@domain1.com
   george@domain99.com
2. Directly edit the users.lst text file for the list and add the email addresses, one per line.

   So to add the same three users, you might edit the users.lst file to look like this:

   u:tam@1.2.3.4 f:Tam Willacy p:0 t:0
   bob@domain1.com
   judy@domain1.com
   george@domain99.com

   where the first line is an existing user on the list.

   Don't worry about the format of lines for existing users. The next time DList has to write anything to the users.lst file it will add the email addresses that you have pasted/typed in correctly.

Adding Users' Real Names

In version 2.5f and above to specifiy the user's real name when you are subscribing them using either method, enter the user's email address with the full name field as per an email client, e.g.

sending subscribe commands:

subscribe "bobby" bob@domain1.com
subscribe "Judy Simpson" judy@domain1.com
subscribe george@domain99.com "Georgie Porgy"

directly in users.lst:

u:tam@1.2.3.4 f:Tam Willacy p:0 t:0
"bobby" bob@domain1.com
"Judy Simpson" judy@domain1.com
george@domain99.com "Georgie Porgy"

When the user subscribes themselves, the real name is taken from their email address (from the From header).

Moderated lists

This is still being written :-)

There are two ways of posting into a moderated list, depending on whether the access_post setting is set to Password or Moderator:

1. Password: DList accepts the message from the user and sends it to the moderator, who if they want to approve it then submits it to the list with the first line of message body being
   approve Password
   e.g.
   approve xxxx

   For this to work the access_post setting for the list in lists.dat must be set to 'Password' and the list password setting must match the password given in the approve line, e.g.
   access_post Password
   password xxxx
2. Moderator: The return address of the incoming message matches that of the moderator setting and the access_post setting is set to Moderator in the lists.dat file, e.g.
   access_post Moderator
   moderator moderator@domain
   where moderator@domain is the address that must match.

Notes:

- There must always be a moderator for a list set to password post access because password is a more secure moderated list.

Archives and Files

This is still being written :-)

DList lists can be set to save an archive of all messages by setting the list specific setting in lists.dat, archive true

If this is set then DList will create the archive messages in a subdirectory called, 'archive' below the list's directory, e.g.
c:\dmail\dlist\listname\archive\1.msg
c:\dmail\dlist\listname\archive\2.msg
etc.

Then if the user sends an email to the listname-request address with the command, dir, in the message body, then DList will send back a message telling the user how many archived messages there are.

If the user wants one of the messages then they can send the 'get' command to fetch the archived message.

If you want to provide other files to the list members, then you create your own directory off the list's directory called, files, and put the files that you want to provide there, e.g.
c:\dmail\dlist\listname\files\picture.jpg

Then when the user does a 'dir' command, they will also be shown a list of other files available.

For details on the list commands see, DList Email Commands

DNAuth - External Authentication Using DNews's users.dat

Under Construction :-)

Sample NetWin Authentication routine, uses DNews's users.dat file.

Version 1.0a - 14 April 1999 - contact tam at support-dmail@netwinsite.com

Included with distribution set for DMail, version 2.5d and above. See the utilities download page to download it separately.

Command line options:
dnauth -check username password
dnauth -lookup username
dnauth -search string
dnauth -path string -... so dnauth uses, string/users.dat and logs there also
dnauth -log -... (or -debug) makes dnauth create log file, dnauth.log in dmail directory.

Normal processing options when used as external authent module for dsmtp/dpop:
exit (or quit)
+OK goodbye

check username[@domain] password fromIPaddress
+OK username@domain drop_file_path uid info

lookup username[@domain]
+OK username@domain drop_file_path uid fwd="a@b" groups="adults"

search string
+DATA user1 info
+DATA user2 info
+DATA user3 info
+OK

Normal processing commands:
check username[@domain] password fromIPaddress
lookup username[@domain]
search string

Works by using a file users.dat in this format
user:password:dnews3:dnews4:dnews5:dnews6:info

NB: colons only allowed in last field, but spaces can be used anywhere. The info has fields, field1="" field2="" where field1 etc. are whatever you want.

DNAuth, takes the information in fields, dnews5 and dnews6 and pretends that they are on the end of the info field as,
name="dnews5" usergroups="dnews6" .
This means that the serch command will find text matching 'string' in any of the fields, user, dnews5,dnews6 and info.

DNAuth will by default look for the dnews.conf configuration file to find the users.dat file. It looks in the path as set in the 'config' setting.

It will also by default, log the last error message to a log file, dnauth.log in the dmail directory.

To make the DMail servers use DNAuth, you should set the following two lines in dmail.conf,
authent_method external
authent_process path_to_dnauth argument
where path_to_dnauth, is the full path to the dnauth process. The argument is optional.

E.g.
authent_method external
authent_process c:\dmail\dnauth.exe -log

Changes History:

Version 1.0b

-

Version 1.0a - 14 April 1999

- added dnews3-6 to User structure
- change rebuild_index to get dnews3-6
- changed free to clear dnews3-6
- checked auth_search only checks user and info fields.
- auth_lookup looks up username and always returns, +OK username config 0
- imported, password encryption from DNews 5.1
- auth_check, made use pass_test
- made str_encode use pass_encrypt
- auth_rebuild, creates name="" and usergroups="" fields so search finds them in info field
- added, quote_check() to check string does not contain quotes, returns, "Invalid Data" if it does.
- made response to quit or exit, '+OK goodbye' as ext auth's are supposed to.
- log pid and timestamp
- log starting message and version.
- debug mode: appends all to log file
  normal mode: log file overwritten with last error
- stops if users.dat cannot be opened in build_index (tries to open twice with 1 second delay)

Products Downloads Prices Support Company

Domains

The whole of the DMail server is based around the email address domain.

All you have to do is tell the DMail servers what email domain or domains you want it to recognise as being 'local'.

You do this using two settings, host_domain and vdomain in The configuration file - dmail.conf.

If you want to support just one domain then you use a host_domain setting.

If you want to support multiple domains then you use a host_domain setting plus a number of vdomain settings.

If you are planning to add many domains, then we suggest that you pick one of these domains to be your 'main' domain.

Use DMSetup to add only this domain, get it working such that you can send and receive email, and only then set about adding the other domains.

The other domains that you add we refer to in this manual as 'virtual domains' although there is very little difference between them and your main domain.

NB: if you don't tell DMail about a domain with a host_domain or vdomain setting then the SMTP server, DSMTP, will think that that domain is not 'local' and try to send on or 'relay' that mail to an outside world server.

On this page...

Related links (not on this page)

- host_domain setting (in reference section)
- vdomain setting (in reference section)
- Should I use username suffixes or multiple IP numbers for virtual domain support? (in Miscellaneous FAQ)

Host Domains

The first (top) host_domain setting found in the dmail.conf file specifies the 'main domain' of the server.

This is the **most important setting** in your dmail.conf configuration file.

This domain should be what the outside work knows your domain as,
e.g. for us our domain is netwinsite.com so on our server we have set,
host_domain netwinsite.com

Your first host_domain therefore must be **resolvable** to an IP address if you are using DSMTP on the internet rather than on an intranet. It is often the A Name in your DNS record.

NB: your first host_domain setting should almost never be your **machine name**, e.g. do **not** make your **first** host_domain setting any of the following,
host_domain mail1.domain.com (**bad** - do not use as **first** host_domain)
host_domain smtp.domain.com (**bad** - do not use as **first** host_domain)

You can set multiple host_domain settings.

The second, third, fourth ... settings are all 'domain aliases' of your main domain. So you **can** enter your **machine name** as one of the lower host_domain settings. Because the domains are aliases, a particular username points to the same user on all of the domains,e.g.,
bob@host_domain1
bob@host_domain2
bob@host_domain3
are all the **same** user, as host_domain1, host_domain2 and host_domain3 are all aliases of each other.

NB: if you add a host_domain setting then in most cases you should ensure that you have also add a DNS entry for that domain to the Internet DNS server system, i.e. your own DNS server or your ISP's DNS server. See the Domain Name Resolution (DNS) section towards the end of this page for details.

For further details on the host_domain setting see the reference section.

If you require the same username to be used for different users on each domain, then you need to setup

what we call 'virtual domains'. Please read on ...

Virtual Domains

Virtual domains allow you to provide email server support for several distinct groups of users who collect their email from supposedly different mail servers.

Such users typically have different domain information in their email address, e.g one group have addresses on the domain, domain1.com and another group have addresses on domain2.com.

DSMTP accomodates these groups by dividing them based on their domain.

The first domain that you set up you should think of as your 'main domain'.

So you pick one of your domains that is going to be the default and call that the main domain. Referring to the section above you add a host_domain setting for it. Then all of the other domains that you add will be virtual domains.

Note that there isn't really any difference between a main domain and a virtual domain, except in how you tell DMail about them.

> (Repeating our message from the overview ...
>
> The setup program DMSetup will only ask you about one domain, so forget about the virtual domains until you have your first main domain set up and tested, i.e. so that you can send and receive mail. Then set about adding the other domains, with the vdomain configuration file setting.)

The biggest difficulty for virtual domains is the POP server login usernames. For example there may be two distinct users both with username fred, let us call them fred1 and fred2

- Fred1 has the email address, fred@domain1.com
- Fred2 has the email address, fred@domain2.com

The DPOP server has to have a method to tell these two users apart when they login, as presumably they both want the login username of simply 'fred'.

DMail provides two ways to do this,

1. **IP Address based domains:**

   Pop users on each domain login to a different IP Address.

   So for example fred@domain1.com might login to ip address 1.1.1.1 with username fred and fred@domain2.com might login to ip address 1.1.1.2 also with username fred.

2. **Suffix based domains:**

   Pop users on each domain login with a different suffix to their username.

   For example, fred@domain1.com logs on to any IP Address with the username,

fred
and fred@domain2.com logs on to any IP Address with the username,
fred@domain2.com

A third method also exists where every login username is unique and you use aliases or a lookup table to create mail addresses. This method is common in products like Sendmail, but it is not recommended by us, see the section,
Unique Usernames Method for Semi Virtual Domains for details.

So how do you add a virtual domain . . .?

You need to manually add (or use the windows GUI DMAdmin) the virtual domains to the configuration file, dmail.conf - see The Configuration File for how to locate and edit this file. The following links to this page take you through the options. (It can be tricky to get your head around so we have provided two explanations of IP based virtual domains.)

1. Adding an IP based Virtual domain - Explanation 1
2. Adding an IP based Virtual domain - Explanation 2
3. Adding a Suffix based Virtual domain

Adding an IP based Virtual domain - Explanation 1

Both domain1.com and domain2.com point to a single machine with multiple IP numbers running one copy of DPOP. In order to handle these two 'virtual-domains' and not mix up fred1 and fred2 even though they have the same username, DPOP must keep them separate. When a connection is made to DPOP it checks the IP number that the connection was made to. This is then matched against entries in the dmail.conf file of the form;

vdomain Btwo 161.33.44.55 mail.domain.one /var/spool/mail/B

vdomain C 222.33.44.22 mail.domain.two /var/spool/mail/domaintwo

The entries in these virtual domain lines are as follows:

vdomain <1> <2> <3> <4>

1. Prefix string used to identify this domain (by default prepended to drop file name)
2. IP number users reading from this domain will connect to
3. domain name which translates to number in part 2 above
4. path for drop files for users of this domain.

These are used to enable the username passed to the authentication process to contain enough information to allow the two users to be distinguished and assigned different drop file paths and passwords.

From the above it might appear that virtual domains can only be supported when an external authentication process is used as two different passwords for user fred are required. This is not the case, as virtual domains can also be supported using normal UNIX user/passwords or Windows passwords by using the following scheme:

- User fred from domain one is 'known' by the DMail servers as A_fred and user fred from domain B is known as B_fred. So their drop files will be a_fred and b_fred respectively (if drop_prefix is true which is the default) and their user database names will also be a_fred and b_fred.

  The usernames and uid's use the form A_fred and B_fred to keep them separate.

  The user does not need to change their email client login username settings as the translation from fred to A_fred is done by DPOP. The vdomain prefix can be a string rather than a single letter, e.g. 'domaina'.
- Connections to IP numbers not listed in vdomain settings, but listed in host_domain settings will use the normal drop_path form and will have normal usernames, dropfile and binfile names.
- The drop file path for for each virtual domain is specified in the vdomain setting
- If a bin path has been specified for bin files this is used for all domains BUT a prefix of A_ B_ etc. is used to distinguish files for users from different virtual domains (If drop_prefix is true which is the default).
- If bin files are stored in the same place as drop files then the drop paths from vdomain settings are used.

Adding an IP based Virtual domain - Explanation 2

Basically to add a virtual domain domain2.com which has IP address 1.1.1.2 you should add a line like,

vdomain dom2 1.1.1.2 domain2.com c:\dmail\in\domain2

In the line above I have chosen a vdomain 'prefix' of dom2, this prefix is used by DMail to refer to users of this domain, so the user bob from the domain, domain2 is referred to by DMail as dom2_bob - never to his face only when it is talking to the authentication programs (a system one or an external one).

The last entry on the line is the path for mail drop files to go for that domain - so that you can keep the mail for each domain separate.

You should notice that all of the entries in the vdomain line, except for the vdomain prefix, are already in your dmail.conf file for your main domain - e.g. the main domain has a setting called drop_path. Note that the main domain does not have a prefix, as usernames without a prefix in your password list are assumed to be for the main domain.

So to add a user bob to your main domain, you simply add the user 'bob' to your authentication database, and to add a user bob to the domain domain2 above you add a user 'dom2_bob' to your authentication database.

Adding a Suffix based Virtual domain

As an alternative to having multiple IP numbers, virtual domains can be set up by asking users of the virtual domain to connect to the POP server with a suffix on the end of their username. E.g. using email usernames of the form Fred/jds.

This form of virtual domains can be an inconvenience to all the users who have to change their client software settings to connect as 'username_suffix' instead of just 'username'.

Mappings from usernames of the form name/domainsuffix are handled by the use of vdomain settings where the IP Address is replaced by a domain suffix. For example we might have:

vdomain J /jds johns.server.domain /var/spool/mail/johnsusers

See IP Address based virtual domains for explanation of the other values of the vdomain setting beside the suffix.

Notes:

- the IP Address is replaced by a suffix which includes the separator character
- The suffix is commonly made, @domain.com so that the username for POP login becomes the user's email address

---

Domain Prefix: drop_prefix setting

This config setting determines how the drop file name is constructed when virtual domains are being used. Drop_prefix can be set to true or false. The default is true. When this setting is true the drop file name will include the virtual domain prefix. For example consider the case when the following vdomain line is being used and a user fred is connecting to abc.com which maps to ip number 1.2.3.4

- vdomain abc 1.2.3.4 abc.com /mail/abc_com

If drop_prefix is true the drop file used will be /mail/abc_com/abc_fred

While if drop_prefix is false the drop file used will be /mail/abc_com/fred

---

Multiple IP numbers on a single machine:

It is fairly easy to add multiple IP numbers for a single machine, up to 255 per interface is fairly straightforward. 1024 is usually possible with minor patches. The exact method varies from one form of UNIX to another - see http://www.nethelp.no/net/vif/readme.html for more information.

As an example, on Linux you would do the following:

```
su - root
ifconfig eth0:2 999.59.4.31 up
```

to add a second ip number 999.59.4.31. The number :2 can be anything between :1 and :255

Common Domain Options

The following discuss some common situations and provide DMail Domain setup options...

1. One domain (-link to Domain Examples section)
2. One domain with a few domain alises (-link to Domain Examples Section)
3. Multiple Domains, Existing User Database of Unique Login Usernames

3. **Multiple Domains, Existing User Database of Unique Login Usernames:**

   Please see the section on this later in this page,
   Unique Usernames Method for Semi Virtual Domains.

Domain Examples

1. One domain
2. One domain with a few domain alises
3. Adding virtual domains to UNIX single domain system

1. **One Domain:**

   Assuming your one domain is, frog.com, then you need to set,
   host_domain frog.com
   in dmail.conf and remove all of the following settings in dmail.conf,
   vdomain
   dpop_host
   any other host_domain lines.

   NB: most people running DMail with one domain have at least their machine name as a domain alias. See the next Example, One domain with a few domain aliases.

2. **One Domain with a few domain aliases:**

   Assuming your one domain is, frog.com, then you need to set,
   host_domain frog.com
   in dmail.conf

   Most people then need to make DMail receive mail addressed to their machine name, e.g. mail1.frog.com

in case someone accidently addresses mail to, for example,
bob@mail1.frog.com
instead of,
bob@frog.com

So you should add a host_domain line to dmail.conf for your machine name.

Ensure that you add it **AFTER** your **FIRST** host_domain line. Using the above domain name examples you should end up with just the following two lines,

host_domain frog.com
host_domain mail1.frog.com

(Remember to remove any vdomain or dpop_host settings in dmail.conf)

Similarly you might need to add aliases for,
pop.frog.com
smtp.frog.com
etc.

To do this, just keep adding host_domain lines onto the end of your list in dmail.conf, i.e.

host_domain frog.com
host_domain mail1.frog.com
host_domain pop.frog.com
host_domain smtp.frog.com

These result in the user bob having ALL of the following addresses,
bob@frog.com
bob@mail1.frog.com
bob@pop.frog.com
bob@smtp.frog.com

Normally bob would only use the first one.

3. **Adding virtual domains to UNIX single domain system:**

   Imagine the scenario below:

   - You currently have a DPOP server running on machine my.server.domain at ip address 100.2.3.1
   - Your email clients connect using addresses such as fred@my.server.domain
   - Their drop files are stored in a directory /var/spool/mail
   - You provide no virtual domain support
   - You use simple UNIX user/passwords for authentication

   Then you are asked to take over (or add) email server support for a second group of users who used to get their email from johns server using addresses such as fred@johns.server.domain What do you need to do to provide a service for this second group:

- Arrange for the DNS to point johns.server.domain at a second IP number assigned to your DPOP machine. Lets say this new IP number is 100.2.3.2 (Setting up multiple IP numbers on a single interface)
- Setup a directory for the drop file to arrive in. Lets say /var/spool/mail/johnsusers for example
- Setup an SMTP server which will deliver mail sent to user@johns.server.domain to this directory. DSMTP could do this for you.
- Choose a letter or short string to prefix users from johns group. Lets say J
- Create usernames and passwords for Johns users. J_fred, J_JSmith etc.
- Add the following line to your file:

  vdomain J 100.2.3.2 johns.server.domain /var/spool/mail/johnsusers
- Restart DPOP

Domain Name Resolution (DNS)

One of DSMTP's main functions as an SMTP server is to send mail which is addressed to non-local users, i.e. outgoing mail from local users and messages being 'relayed' through DSMTP.

Before DSMTP can send mail out it has to resolve the domain name in the destination email address to the IP address of the destination email server.

In normal operation it does this 'resolving' by sending a lookup request for the domain to a DNS server.

So to answer the question, **"Do I need a DNS entry for the DMAIL server?" ...**

Yes, you need to set up either an Mail eXchange (MX) DNS entry for your domain, OR and 'A' or 'C name' DNS entry.

Your service provider should be able to help you with this if you do not run your own DNS server. Feel free to ask DMail Support, if you want to clarify anything on this before talking to your service provider.

The use of the DNS server can be by-passed by setting a gateway setting for the domain in dmail.conf.

By default DSMTP will use the DNS lookup procedures setup in your operating system for your machine - so basically if you can ping a domain then DSMTP should be able to resolve that IP address too. Your operating system probably uses a list of DNS servers for domain to IP address resolution as well as the 'hosts' file for resolution of some local domains, e.g. /etc/hosts on UNIX systems and \winnt\system32\drivers\etc\hosts on Windows NT.

If you want to specify a particular list of DNS servers for DSMTP to use then you can enter multiple, dns_host settings in dmail.conf. These will be used by DSMTP **instead** of any system DNS lists or hosts file.

DSMTP caches the last 1000 DNS lookups that it has done and clears this cache every 2 hours or when the server is restarted.

DNS Caching to Disk - Unique New Feature in version 2.8i

Many DNS servers on the internet have a bug where they incorrectly report a domain as unresolvable when a short time before they could resolve that domain. So we have added a unique feature to DSMTP. By default it now caches to disk successfull DNS lookups, and checks that cache if a DNS server reports a domain as unresolvable.

This does mean that if someone removes a domain then DSMTP will continue to try to deliver to that domain's old IP Address. However this is not a common situation whereas the very real problems caused by false 'invalid domain' responses can cause a huge administration problem.

The DNS caching feature can be disabled with the setting, dns_disk_disable true.

What order does DSMTP do lookups? (MX Priorities)

DSMTP will first do an MX lookup and use any IP addresses that it finds in the priority order in which they are given, if none of those are successful then the message is queued for a later retry.

If the MX lookup fails then DSMTP will do a DNS (A) lookup for a straight IP address and use it if it finds one. If it does not find one then it will bounce the message.

So if there is a record in either lookup then DSMTP will not bounce the message, even if no SMTP server responds at the IP address that it has found. Instead it will queue the message and retry to send it later. There is a limit to the number of retries that it does, after which it will bounce the message. This limit is set by
max_retrytime x
where x is in hours and is the length of time it keeps retrying for.
The time between retries is fixed at 2 hours, but you can use the
tellsmtp tryall
command to initiate a retry for all queued messages if you have a backlog to clear.

Do I have to register a URL for the DMail server?

If you are asking this question then we presume that you are planning to run one of our Web to email gateway products, e.g. CWMail which is a CGI, in order to provide web access to email on the DMail server.

These require that you set them up on a web server.

Yes you do need to register a DNS entry for your domain, e.g. supposing you wanted dmail to handle mail addresses with the domain, big.com you might have an A or main record for,
big.com --> ip_address_of_dmail

(this would match your first host_domain setting in dmail.conf)

Such an entry would allow URLs starting,

http://big.com/...
to reach the web server on your machine.

Commonly people then add a 'C name' or alias DNS entry for www.domain.com, so continuing the example above maybe,
www.big.com --> big.com --> DMail_ip_address which would then make URLs starting, http://www.big.com/... reach your web server.

Upper Case Domains

Talk to us if you want them - but basically DSMTP will always look for the lower case form of the domain in settings in dmail.conf.

So if a message comes in for bob@Domain99.com, then DSMTP will look for a host_domain or vdomain setting for the domain, domain99.com NOT Domain99.com. Similarly all forward rules etc should specify the domain in a lower case form.

Unique Usernames Method for Semi Virtual Domains

- **Multiple Domains, Existing User Database of Unique Login Usernames:**

When moving to DMail you might have already have a database of unique usernames for users across multiple domains. E.g. you may have the the addresses,
bob@domain1.com
bob@domain2.com

These two addresses deliver to two users, bob on domain1.com who has the login username,
bob and bob on domain2.com who has the pop login username of,
bob2
(or some other unique username, e.g. xyz.fred)

Answer:

Basically such a system does not tie a username to a domain.

DMail's virtual domains **do** tie a username to a specific domain.

In order to take advantage of a number of features you will have to change your domains to DMail's style of virtual domains.

We recommend that you take this opportunity to change, as there will probably be future features which you will want to take advantage of, which require that you have DMail style virtual domains. Ways in which you can change are detailed in option 1 below.

However it is possible to setup DMail to continue functioning with your user database in the format that it is currently in. Options 2 and 3 below cover this.

You may also want to setup DMail to continue functioning with your old user database for your existing domains, but then add DMail style virtual domains for any domains that you add in the future or even move your existing domains over one at a time at a later date.

Here then are all of the options for dealing with this situation:
(notes and our recommendations follow this list)

1. make the two domains separate virtual domains ...

   1a. by moving each of the domains onto separate IP addresses.

   1b. by making bob on the second domain loog in using a suffix , e.g. his login username changes from bob2 to bob@domain2.com (i.e. domain2.com becomes a vdomain)

   1c. by making both bobs login using a suffix, e.g. bob changes to a username of bob@domain1.com, and bob2 changes to a login of bob@domain2.com

2. make the two domains aliases of each other with two host_domain settings,
   host_domain domain1.com
   host_domain domain2.com
   authent_domain false
and add an address alias that points,
   bob@domain2.com to bob2@domain1.com

   2a. create alias for bob2 in alias_file_domain setting

   2b. create alias for bob2 in fwd="" field of user lookup response.

3. create aliases without having to add host_domain settings for second (or further domains)

   3a. use virt_user_pre setting to create a Sendmail style 'virtusertable', where, bob@domain2.com redirects to bob2(@domain1.com).

   3b. use forward settings in dmail.conf to redirect bob@domain2.cmo to bob2(@domain1.com).

Notes:

1. Options 2 and 3 mean that you cannot later add a user, bob2@domain1.com, as that address belongs to bob2 on domain2.com even if he does not use it.

   This is probably not an issue if your users currently use wierd login usernames that are not likely to be wanted to be repeated e.g. xyz.fred. There is also the option to make your first host_domain a dummy domain that is not actually used in email addresses.
2. Options 1b and 1c, can be made more bearable if the users will be using CWMail (or any other of the Web to email gateways) to read mail, because they can automatically add the suffix for the user on login to the web site.

Our Recommendations (in order)

1. If you have excess ip addresses ...

   definitely choose option 1a, because you can run our style of domains without having to change

the login username.

2. If you are willing to force users to change their login username ...

   choose option 1b, because only the users on domain2.com have to change their login username.

3. If you are **not** willing to force users to change their login username ...

   choose option 2b, because it allows you to administer aliases in the user database, which also means that you will be able to alter them using the Web based user admin tool, NetAuth.

   Also remember that if you add further domains in the future then you should try to add suffix based virtual domains.

Notes on updates to DMail, comprising the DSMTP, DPOP and DList mail servers

(in reverse order)

For newer updates information please see updates.htm

Version	Date	Changes
2.2p	10 Aug 98	• tellsmtp profile • dmail.conf setting lock_id nnn, this can be used when running multiple coppies of dsmtp/dpop over NFS volumes, this makes dmail use an internal locking mechanism that will work over NFS.
2.2j	27 July 98	• Added auth_nocache true option for dpop to disable caching of passwords (This only applies to /etc/passwd entries) • Added nwauth.exe, nwauth.c, nwauth.h, wadduser.c example external user authentication module and web interface for adding and modifying users • Modified hashing method used for security, this means if you have tellnews or dmadmin for an earlier version you must update it to match your new version.
2.1h	28th Apr 98	• alias_file_domain parameter added (DSMTP) • forward_from parameter obsoleted (DSMTP) • forward_from_ip parameter added (DSMTP) • nameserver info extracted from resolve.conf or registry if no dns_host found (DSMTP) • fixed log rotate problem introduced in 2.1a • Tellpop drop command timeout increased for large drop files • fixed spelling of vdomain_separator msg_separator config settings - either spelling accepted. • new spawn routines for NT so external authentication works when running as a service

2.1d	7th Mar 98	• $user$ template string allowed in drop_path setting • check_owner_disable setting added to disable drop file owner checking on Unix • bug causing occasional "key for wrong PC" when loading new key fixed.
2.00m	5th Mar	• fixed timeout bug on Tellpop sendlog, shutdown • for spool_hash 1 and 2 DPOP now creates the intermediate directories as required
2.00j	12th Feb	• fixed bug in Tellpop reload with large numbers of virtual domains
2.00i	12th Feb 98	• first beta release of full DMail package • bulletin board system added • manual updated • manager password stored encrypted in separate file • manager password generally not passed as clear text across net • no more gratuitous messages on test versions • X-DPOP header on first message of burst on trial versions • Tellpop bulletin command to create simple bulletins • Tellpop conf lists current config settings • vdomain bug fixed • max vdomains increased to 512 • tellpop sendlog sends all logged info messages • drop_prefix true/false config setting added
1.19d		• Max length of username + virtual domain prefix increased to 80 characters • Max path length for Unix increased to 128 characters • bug in drop_all command fixed • rotate 4 old log files called dpop1.log, dpop2.log etc.
1.17c	20th Dec	• bug in slave chan burst with blank user fixed. • bug in virtual domain handling fixed • change to use dmail.init for startup rather than rc.local • work around for bug in select call on some machine which caused occasional excessive CPU usage.
1.15		• If a drop file is locked on connection DPOP will wait (non blocking) rather than rejecting connection.

1.13	11th Nov	• Drop file lines > 5,000 are now wrapped - were killing DPOP • Tellpop on Linux is put in /usr/bin rather than /usr/local/bin as /local is not in default path anyway
1.12b	18th Nov	• fix for bug in freeBSD and BSDI versions causing invalid file handle errors.
1.12	9th Nov	• Added lowercase_username and lowercase_password config settings to allow case insensitive user/password • bug causing problems in retrieval of some large messages fixed.
1.11	4th Nov	• tellpop abort fred command added • tellpop list_current extended • bug causing chans not to timeout in some circumstances fixed
1.10	31st Oct	• Virtual domain prefixes can be any string rather than single letter
1.09	29th Oct	• Intermittent slowdown on Linux fixed • Bug causing occasional crash on some systems fixed
1.08x		• default for slave_process did not include path - sometimes not found • minor bug fixes
1.08w	21st Oct	• Null passwords caused crash - fixed • Possible loss of chars on large RETR - fixed • Don't start first slave or authent process unless required.
1.08u	15th Oct	• slave_burst_size default was 0 instead of 1000000 • slave_timeout setting in confif file was ignored and random value used.
1.08t	15th Oct	• Faster processing of connections with no new messages • optional slave processes for burst of large drop files • more status information displayed • status info logged periodically • retr loop bug on slow links fixed • slow retrieve of large messages fixed • faulty bin count causeing slow down with uptime fixed. • allow usernames/passwords which include space characters. • dont timeout on retrieve of large messages on slow connections • limit chans to actual available file handles. • enable/disable individual users

		• tellpop listcurrent to give current connections • store/display reasons for going offline • shutdown n to allow connections to exit nicely if possible.
1.08g	25th Sept	• Support for virtual domains added • various bug fixes
1.07i	22nd Sept	• Stats report tidied up • die command added • corrupted/aborted stats files dont kill tellpop anymore
1.07h	18th Sept	• Corrupted or lost idx files are regenerated when possible. • Possible death on 100+ sessions fixed. • Death from broken pipe fixed. • File leak from external authentication fixed. • Numerous minor bug fixes
1.07e	15th Sept	• Default timeout for pop chans 10mins rather than 1min • Timeout is now from last transfer not just last command
1.07d	11th Sept	• Bug in POP3 LAST command fixed
1.07c	10th Sept	• Limit max connections to 1/3 available file handles as need average of 3 per connection. Particularly relevant on linux which defaults to 256 file handles and 200 connections.
1.07b	10th Sept	• Increase file handle limit to allow more concurrent connections. Limit on solaris/bsdi was 64
1.07a	10th Sept	• Bug fix: error on chan caused crash • 100+ concurrent sessions caused offline • max_sessions exceeded caused offline
1.07	8th Sept	• Minor bug fixes in external authentication routines.
1.06b	5th Sept	• minor bug fixes.
1.06	27th August	• pop port timeout implemented • non existent directory for .bin files could cause cause crash - fixed
1.05	21st August	• Added command tellpop stats which analyses a wildcard set of .stats files and gives per user information for accounting etc. • Added command tellpop help

1.04	14th August	• drop_users setting added to dpop.conf to allow for a few users who need to read mail from Unix command line AND via POP3 from client software AND leave messages on the server. • Compact and drop VERY large messages without usig lots of memory • Default location for work_path changed from drop_path to dpop_path • Bug fix: External authent process which timed out could be left in zombie state on unix • minor bug fixes • check_gid setting allows for mail group called something other than mail • dp_setup allows mail user/group with name other than mail
1.03	5th August	• Bug fix - some versions of linux rejected users if shadow password was not being used. • First solarisx86 version.
1.02	3rd August	• Config option to process .username.pop and other temporary files from previous poppers. • dp_setup wizard upgraded. • DPOP Removal section added to manual
1.01	1st August	• Option of drop files with gid=mail or gid matching user uid and no group access. • dpop files set to owner = users uid and users group and no group access, user read-only
1.00	1st August	• First full release. • Added online/offline states for safer shutdown and removal of DPOP • drop_all command now supported with external authenticator
0.68	24 July	• Support for Mailbox format added • More status in tellpop status • Bug in included From line handling fixed • Extended external authenticator handling • dp_setup wizard remove option extended.
0.67	18 July	• Bug in chan close by remote end fixed - only affects solaris. • Duplicate user rejection bug fixed

0.66	16 July 1997	• Fixed bug when using .\n as msg seperator • changed external authenticator to include drop filename in drop path
0.65	July 1997	• First Solaris beta version • Various minor bug fixes
0.64	July 1997	• NT User password checking implemented • various bug fixes
0.62	July 1997	Various updates to manual and dp_setup wizard
0.61	July 1997	First beta release of DPOP for Linux

Products Downloads Prices Support Company

Disk Use And Files

An incoming message for a user is appended to that users unique 'drop file' by DSMTP. DSMTP will create the file if necessary. Then when the user connects to DPOP to read their mail, DPOP 'bursts' open that drop file and adds the messages that it finds to its own indexed 'bin' files. Then DPOP feeds the mail messages to the email client as it requests them.

DPOP can be made to 'drop' its bin files back to the drop file format for compatibility with other email servers, but this can affect performance.

See the list of DMail files at the bottom of this page.

Drop Files

Drop files are the link between SMTP and POP servers.

DMail creates and uses the Unix 'standard' drop files as used by Sendmail etc.

The drop file is a text file containing all the user's email messages one after another, including all of the message headers. Each message is separated in the file by a blank line followed by a line, starting 'From '. DMSTP inserts the From line at the start of each message (DPOP will also insert such a line, when it is asked to drop its bin files, back to the drop file format).

E.g. Example drop file

From pntest@161.29.2.46 Sat Dec 26 09:37:48 1998
Return-Path:
Received: from tosh ([161.29.2.46]) by tosh ; Sat, 26 Dec 1998 09:37:48 +1200
Comments: Authenticated sender is
From: pntest@161.29.2.46
To: Postmaster@161.29.2.46
Date: Sat, 26 Dec 1998 09:37:47 +0000
Subject: bob
Message-ID: <91461826801@tosh>

message body

the end.

```
From tam6@[161.29.2.46] Tue Jan 05 16:41:58 1999
Return-Path:
Received: from tosh ([161.29.2.46]) by tosh ; Tue, 05 Jan 1999 16:41:57 +1200
Comments: Authenticated sender is
From: "Tam Willacy"
To: tam@iba.com
Date: Tue, 5 Jan 1999 16:41:34 +0000
Subject: thanks
Message-ID: <91550771801@tosh>

Thanks for buying DMail :-) :-)

**************************************************
Tam Willacy,Product Development
Netwin Ltd.
support: support-dmail@netwinsite.com
email: tam@netwinsite.com
**************************************************
```

Note: DPOP does not leave messages in the drop file format, so once a user has checked their mail box, their drop file will be left empty. See the section below on bin files for more information.

The Drop File Filename

The name for each user's drop file (or mailbox) is based on the username that they log in with.

So if bob is a valid user then his drop file is called simply, 'bob' (note that there is no DOS file extension).

There is just one other option for the name of the drop file.

Users on virtual domains by default have a prefix added to their drop file filename so that users mail does not get mixed up even if they happen to be stored in the same directory. The prefix is one of the things set by the vdomain line in dmail.conf

The prefix is attached to the user's name with a special character which by default is an underscore, '_'. In general NO USER on your system is allowed this character in their username. This character can be set to something else with the setting, vdomain_separator

As an example bob from domainx.com might end up with a drop file name of,
domx_bob

See the section below on the drop_prefix setting for more on this.

Often with virtual domains there are fancy schemes where users have to log in with strange usernames - e.g. in suffix based virtual domains they have to log in with a suffix to show that the user belongs to a

particular domain (bob@domainx.com or bob/domainx.com). **NOTE** in DPOP these **never** carry through to the drop file name.

There are only the two options for the name of the actual drop file.

So even though bob might log in with bob@domainx.com because he belongs to the domain, domainx.com, if DPOP recognises that domainx.com is a valid virtual domain suffix then his drop file will be called simply, 'domx_bob' or just 'bob'.

---

drop_prefix

The setting drop_prefix specifies if the prefix for a virtual domain is put on the drop file name. The prefix allows you to tell which drop file belongs to which domain. It also allows you to place mail from different domains in the same directory. The prefix stops users with the same name from having the same drop filename, e.g dom1_bob and dom2_bob.

By default it is true, as it is a good safety measure to have drop files named differently.

Bin Files

This is the format that DPOP stores user's mail in while it still resides on the POP server. This is generally only until the user reads (retrieves) their email, with their email client.

The user's bin files are created in a directory of the same base name and location as the user's drop file, but with the .bin extension added to its name.

E.g. a user Lucy whose drop file is,
c:\dmail\in\domain1\lucy
would have the bin file directory,
c:\dmail\in\domain1\lucy.bin

Within that directory DPOP creates message data files,
bin00000.dat
bin00001.dat
. . .
and a couple of index files, bin.idx msg.idx

. . .

The dropping of bin files back to drop files by DPOP can be initiated manually e.g.
tellpop drop username
or
tellpop drop_all

or it can be done automatically after a user disconnects from reading their mail, by setting the dmail.conf setting,
drop_users <wild card list>

Queue Files

All messages that arrive at DSMTP's door get put in the work_path directory as a queue file. Then other sections of dsmtp deliver them either locally or non-locally or to robots etc.

These files are in pairs - q_xxxx.itm being the message and q_xxxx.idx being an index of the envelope etc. You can open these up with a text editor. The .idx file is particularly useful as it contains information on the number of times a message has been tried etc.

Note: there is one pair of queue files per message in the queue but each message may have a number of final destinations. The idx file shows a 'fwd' line for each destination address (recipient line) with information on when it was last tried and if it was successful or not. Each queue file remains until all recipient lines for that message have been dealt with.

The way that DSMTP deals with these messages is complex. But basically it loads up 1000 of the recipient lines from the queue files into internal memory and then sets about delivering to each of those destinations. It does things like scanning the internal queue reordering it, moving messages on and off it, looking for destinations that are at the same domain to send down a channel it already has open, etc.

The tellsmtp status command lists amongst other things the number of queue files with its 'pending' counter. It also lists those messages which are in its internal queue (see max_queue). The numbers down the left hand column equate to the queue file number.

If DSMTP cannot deal with the current load then the number of queue files can back up. This is not a bad thing - it is the way SMTP servers work! When things quieten down a bit those messages will get delivered. What is a bad thing is if it backs up so much that messages never get out of the queue or over a period of days the queue continues to grow, i.e. it does not level out.

Typical ballpark queue sizes are,
100 for less than 1000 users
1000 for 1,000-500,000 users.
If your queue is often over these guidelines then you should check that that is ok for your system - what counts is throughput, i.e. are your messages getting through in an amount of time that you are happy with.

Note things like large mailing lists can greatly affect queue sizes.

Being hit by a spammer is a common reason for your queue to shoot through the roof! Using utilities like grep or find in the work_path directory you can often track down messages with the same sender or subject line. See the Spam section for details on how to stop the spam.

Talk to DMail Support if you are worried about a queue problem.

See also this FAQ, Can I delete queue files from the queue?.

Drop Paths

The path for your mail drop files is set for your main domain with the dmail.conf setting, drop_path.

For each virtual domain the last value in the vdomain setting sets the drop path, e.g.
vdomain dom1 1.2.3.4 domainone.com \dmail\in\domone
sets the drop path for the virtual domain domainone.com to \dmail\in\domone.

If you have a number of synonyms for your main domain, i.e. a number of host_domain settings, then you can specify that certain domains put their mail in specific directories with alt_drop_path settings. NOTE that you cannot use alt_drop_path when using external authentication or for virtual domains.

The drop path can also be extended by using directory hashing> as detailed in the next section.

Hashing

In email systems with a large number of users you can easily end up with an unusably large mail directory. Worse is that performance can be severely affected by 'large' directories, particularly on UNIX based platforms.

To get around this dmail offers two forms of directory hashing. You turn these on with the hash_spool setting. Simply, hash_spool 1 adds one extra level of directories named with single letters of the alphabet, spreading mail drop files across 26 directories. Hash_spool option 2 adds two levels and so 26 squared directories are created.

All hashing directories are ADDED to the 'base' mail location set by the drop_path or vdomain settings.

Note: DSMTP creates these directories as mail arrives to be put in them. DPOP will also create them if it needs to for any reason.

Hashing is not done on virtual domain prefixes. See the reference section for more details on hash_spool

---

Quotas and Message Size Limits

Relevant settings: max_msgsize, drop_max, user_quota

max_msgsize limits the size of any mail message that passes through DSMTP, whether for delivery to a local user or being sent out.

drop_max limits the size of a user's drop file. If DSMTP is trying to deliver a message to a local user it checks to see that the user's drop file isn't going to be bigger than this limit after it has added the message it wants to add. If it is then it bounces the message and notifies the sender that the recipient was not able to receive the message.

Note that when running DPOP a user's drop file is often 0 bytes in size. This is because when the user connects to check their mail, DPOP clears the drop file. DPOP sorts the messages from the drop_file into its own BIN files.

So to limit the users' disk usage it is advisable to set up a quota system.

Turning on the Quota System

To do this, in dmail.conf you need to add the setting,
user_quota true
then for each user who you wish to have a quota limit, you need to go to the directory where their 'drop file' is, and create a file called,
username_inf
where username is the user's username as it appears for their drop file (note that it may a have virtual domain prefix on it).

In the username_inf text file you need to add a line like,
quota 40000
(to set a quota of 40 kbytes)

DPOP maintains a line,
used xxxx
in the same file, so that when DSMTP receives a message for that user it will check to see that the used amount plus the size of the new message will not exceed the quota limit.

In version 2.4i and above you can set a default quota for all users by simply giving a default quota in kbytes in the user_quota setting, instead of the word 'true', e.g.
user_quota 40
sets a default quota of 40 kybtes for all users that do not have a quota line in their username_inf file.

Remember that in the _inf file the quota setting is in BYTES, but in dmail.conf the user_quota setting is in KBYTES.

NB: The Web Gateway products, CWMail and DMailWeb etc., have their own quota systems which limit the amount of space a user uses on the web server for storage of read mail etc. The DMail quotas only affect space each user has in messages waiting to be read and messages 'left on the POP server'.

Path Settings

. . . drop_path, alt_drop_path, work_path, dlist_path, log_path, dwatch_path

Server Farming, NFS

Relevant setting: lock_id nnn

The servers in DMail are quite happy for you to put the users' drop and bin files on a shared network disk.

So you can also run multiple DSMTP and/or DPOP servers in order to share server load or for redundancy. You simply set the relevant path settings to a shared network drive, and then set a unique lock_id on each machine.

Basically the servers work in the following way:

Users connecting to read mail can only connect on one POP server at a time, so when a user logs in DPOP locks the user's drop file to read it, closes it and then talks to the user. If it finds the drop file already locked then it won't let the user connect.

Mail can come in for the same user on all servers in the farm at the same time, so if DSMTP finds that it cannot access the user's drop file, then it retries to access it 3 times at 1 second intervals.

On Windows platforms, you need to use UNC style names (e.g. \\serverA\shared_drive\dmail) to specify paths for the settings above between the server machines for the mail storage area. You may also need to use a UNC name for the authentication module, e.g. our NWAuth module requires this. You can map network drives instead of using UNC names but UNC names have the advantage that if the server reboots then the path specified with the UNC name will be accessible, whereas a mapped network drive is not accessible until someone logs on and the mapping is created. Whether you use UNC names or mapped network drives you MUST enable sharing on the folders that the servers are to share AND ensure that the correct permission for that shared resource is given - see the note below.

NB: the DMail servers are spawned by the DWatch service on NT, which will be running as the user 'system', so that is the user that the servers and the authentication module will also be running as. You cannot give access permission to a mapped network drive for the 'system' account. So you must change the NT DWatch service so that it runs as a specific user (in Control Panel, Services, Startup) and then give read/write permission for that user on the shared directories.

On UNIX based platforms: DMail and NFS drives

DMail has been designed to work with Network File System (NFS) drives. They can be used to allow DSMTP or DPOP servers to share directories such as the mail spool area (drop_path) and drop_path

NFS can cause a lot of problems for mail servers because of its lack of file locking and delayed file updates. When you use the following setting DMail will use additional locking mechanisms to avoid problems with NFS

Set in dmail.conf lock_id to a unique string for each server that shares the mail spool on the NFS drive.

Please read the following list of notes if you are using or intending to use an NFS drive with DMail.

Notes on DMail and NFS:

- **Use at least version 2.8i:** We have recently improved our NFS file locking code. Version 2.8i has improvements for DSMTP and improvements for DPOP will follow in in 2.8j
- **NWAuth:** NWAuth can be used by multiple DSMTP and DPOP servers at once on an NFS drive. However it does no locking at present so should not be run in a situation where multiple copies will add a user at once when it is being run on an NFS drive. This will be fixed shortly. Contact DMail Support for further information.

Log Files

log_path . . .

Log Files: Logging of information and error messages

Follow this link for help on Deciphering Log Files.
'ded' files

Various informational messages are logged during the normal operation of the DSMTP, DPOP and DList servers. These events logged include such things as server startup, creation of work files, rejection of connection request etc. These messages appear in separate log files for each application; dpop.log for DPOP, dsmtp.log for DSMTP and dlist.log for DList

When the log file exceeds a preset maximum size it is renamed and a new file is created. For example dpop.log becomes dpop1.log This is referred to as log file rotation. For DPOP four old log files are kept - dpop4.log is the oldest and dpop.log is the current log file.

The path for the log file is normally the same as the work directory but can also be set individually in the configuration file using the log_path setting. For Example:

log_path \data\logs

In addition, error, warning and debug messages can optionally be stored in the log file. The level of information logged is set by the log_level setting in the configuration file. The available settings are shown in the table below:

error	only logs error messages
info	logs informational messages and error messages
debug	logs debugging messages, informational messages and error messages

The default setting is **info**. The log_level setting may also be changed temporarily using the Tellpop or Tellsmtp log_level commands:

e.g. **tellpop log_level info**

The logging level will revert when the system is restarted or when the configuration file is reloaded.

When one of the servers dies, DWatch should notice and restart the server (see the dwatch section for more details on when it does not restart the server). When it does this, by default it will RENAME the current log file for that server, e.g. dpop.log, to a 'ded' file, e.g. d_1dpop.ded. These ded files are named as, d_xserver.ded, where x is incremented on each death (so 1 is the first death dwatch caught), and 'server' is the name of the server, i.e. dpop, dsmtp or dlist.

Statistics Files: Gathering connection statistics

DPOP and DSMTP can provide various statistical information. In particular DPOP can optionally provide a per connection log of client usage statistics for use in charging clients etc.

These are the log/statistics files that are produced in addition to the standard log files.

- Tellpop status command - see tellpop commands
- Tellsmtp status command - see tellsmtp commands
- **daily dpopmmmdd.stats files for dpop (dpopjan01.stats):**

  A new statistics file is produced each day with a filename dpopmmmdd.stats which is intended for billing purposes.

  For example for the first of January the file name will be dpopjan01.stats

  The statistics file is a printable text file containing one line per connection with the six fields separated by spaces:

  - start time of connection in seconds
  - end time of connection in seconds
  - username
  - from host name
  - bytes transferred
  - number of messages transferred

  To disable DPOP's per connection log modify the configuration file to include a blank path for the stats_path:

  Example:

  stats_path my/stats/path
  becomes

stats_path
To disable logging of connection statistics for charging

- **a tellpop command to collate dpop stats for a monthly period:**

  The command: tellpop stats can be used to summarize the information from a collection of .stats files. It will produce a one line summary for each user giving number of connections, number of messages, average time between connections etc. See Tellpop commands section for more details.

- **daily .sta files for dsmtp (dm0101.sta):**

  This file contains the output of a tellsmtp status command and is created at the end of each day.

- **daily dmddmm.log files for dsmtp (dm0101.log)**

  This log file contains a simple log of incoming/outgoing messages and whether they were delivered. It builds up throughout the day and gets rotated at the end of the day

- **A stats dump file,dmddmm.dmp (dm2712.dmp, 2.7h and above):**

  This file is appended to throughout the day if you add the setting,
  dump_stats true
  to the dmail.conf log file.

  This file is intended to help support to work out what is happening on your server over the day, e.g. to pick up on DNS connnection problems and periods when queue files get backed up.

File Permissions

. . . mostly as MAIL user and group

drop files will be set to 600 on Unix platforms if not already

DPOP: check_gid =

check_disable true = be happy with file permissions and don't change them.

DSMTP: chmod_drop false = don't change drop file permissions, but still create on Unix platforms with 600 file permissions. So doesn't affect things like a forwarding alias to /dev/null/ (note forwarding to /dev/null can also be achieved by making an alias to user@nul, upon which DSMTP will make it the message disappear into the void :-)

List of DMail files

This list is not complete but files that we are commonly asked about will be added here as links to the section describing them.

ded files	(typically in \dmail\dwatch or /usr/local/dmail/dwatch) d_1dpop.ded d_2dpop.ded d_5dlist.ded	when dwatch has to restart a server it archives the current log file, e.g. dpop.log, by renaming it to a 'ded' file, e.g. d_5dpop.ded is the log file on the fifth death of dpop.
exit_files	(typically in \dmail\dwatch or /usr/local/dmail/dwatch) dpop.exit dlist.exit dsmtp.exit	any file named this in the dwatch directory will cause the named server to shut itself down.
join.tpl - DLIST	(typically in \dmail\dlist\list_name or /usr/local/dmail/dlist/list_name) join.tpl - can go in dlist directory(applies to all lists) or in individual	This file has a Welcome message for new subscribers to either a particular list or all lists depending on its location. It is a template of the message - dlist fills in list name etc for you.
q_x.idx and q_x.itm - DSMTP	(typically in \dmail\work or /usr/local/dmail/work) q_xxx.itm - queued message xxx's data, q_xxx.idx - DSMTP's index for the same message	This file has a Welcome message for

---

Deciphering Log Files

Under Construction :-)

Firstly it should be noted that in general the log files are there for when things go wrong. As such they need to contain enough specific information that DMail support staff can track down what has gone wrong. This unfortunately is what tends to make them unreadable :-)

This page has a few pointers and a few strings that are useful for working your way through a DMail server's log file, looking to see if a message has been delivered etc.

The daily summary logs that DSMTP creates, i.e. dmddmm.log, e.g. dm0304.log, are far more useful for finding out this sort of information, but we appreciate that at times the information in the log file is very useful to System Administrators as well.

The other useful tool is the status commands for both tellsmtp and tellpop. Probably the biggest thing to watch is the pending count. A rough guide is, on a small system in the 10s, medium system 100s, large system maybe into the thousands. A better guide is to take note of what is normal for your system and if it increases by an order of magnitude . . . panic :-)

Searching for:

Searching for failed deliveries

Search dsmtp.log (or dsmtp1.log, etc. for older messages) for the string '(failed)'

It will be on the end of a line like,
** Could not deliver message from <bob@domain1.com> to <julie@domain2.com> (failed)

This will only have been logged if you were running DSMTP on info log level.

You will need to search upwards in the log from this line to find out the reason for the failed delivery.

Example 1 - TCPIP or Socket error

> 18/03 10:36:45 *** Warning *** sock: (Error on channel) The virtual
> circuit was reset by the remote side.

This is a normal glitch in the TCPIP protocol.

> 18/03 10:36:45 *** Error *** In tcp_write (error) cant write 220
> domainx.com DSMTP ESMTP Server v2.4f

The consequence of the above.

> 18/03 10:38:16 ** Lookup domain for channel 0 is domainx.com
>
> 18/03 10:38:24 *** Error *** tcp: Channel closed or didn't open [2] 156

Means that the other end of the TCPIP channel did not respond, the server may be temporarily down. The line above it is probably not related. Again it is almost certainly nothing to worry about.

> 12/01 13:39:37 *** Warning *** socket: EINPROGRESS The open is pending

This is just TCPIP level socket information which we accidently made a warning message. It is nothing to worry about and in later versions (2.7n I think) it becomes a debug level only message.

DWatch Utility

When DSMTP, DPOP and DList are installed a small but very important utility, DWatch, is normally also installed. It starts up the three servers at boot time and then periodically checks they are still running. If for some reason they have died abnormally then DWatch will restart them, and optionally archive their log file for later analysis (a 'ded' file).

NB1: On Windows NT dwatch is run as a service and in turn it starts DPOP, DSMTP and DList. So to stop the servers when DMAdmin is not working you should stop the dwatch service from NT's Control Panel, Services window (see also exit_files below).

NB2: On Windows NT if one of the servers dies then by default a system alert window will pop up. When this happens dwatch cannot detect that the program has died until you manually close the alert window. So to allow dwatch to restart the server when it dies you should go to the 'DWatch + remote' tab in DMAdmin, click on the 'Config DWatch' button, and then click on the button marked 'Don't pop up dialogue when any program dies' under ANY one of the servers being monitored (you can also set the debugger to DrWatson).

The use of DWatch is controlled by the use of a DWatch directory, set by the config setting dwatch_path. DWatch looks for files in this directory called xxxx.wat containing information on how to restart xxx, how often to check that it's running and how many times to restart it.

The xxx.wat files are normally created by the DMSetup wizard, from then on they are easily modified in DMAdmin, or directly in a text editor, e.g. notepad or vi.

`Here is an example .wat file for DList:`

exec C:\dmail\dlist\dlist.exe	# specifies full path to executable file
time 20	# delay between 'still operating' check, in seconds
rest 5	# number of times to retry starting program
logfile C:\dmail\log\dlist.log	# full path to log file for the program being watched
pidf C:\dmail\dwatch\dlist.pid	# name of pid file (with full path) to check

The servers themselves write xxx.pid files (process id files) to the DWatch directory when they are started and remove them on normal shutdown.

The default dwatch_path setting is dpop_path, however on Windows NT the DMSetup installation wizard creates a specific 'dwatch' directory as default and points dwatch_path at it.

```
Example dwatch_path setting in dmail.conf:
```

UNIX based platforms:
```
dwatch_path /usr/local/dmail/dwatch
```

Windows platforms:
```
dwatch_path \dmail\dwatch
```

DWatch also creates a temporary log file and stores it at the path given by the log_path setting (the same as work_path by default). This log file contains only DWatch information and is overwritten at the start of each DWatch session, i.e. on restart or at the start of the following day.

Ded Files

When one of the servers dies, DWatch should notice and restart the server (see the note NB2 above). When it does this, by default it will RENAME the current log file for that server, e.g. dpop.log, to a 'ded' file, e.g. d_1dpop.ded. These ded files are named as, d_xserver.ded, where x is incremented on each death (so 1 is the first death dwatch caught), and 'server' is the name of the server, i.e. dpop, dsmtp or dlist.

Ded files are created in the dwatch_path directory.

Exit Files

A useful trick to know is that you can stop most of the servers by putting a file in this directory named, server.exit, e.g. if dlist sees a file called dlist.exit in the dwatch directory, then it will shut itself down (and remove the exit file when it does so).

How to set up a hotmail system.

This guide will take you through the setup of a "hotmail" system, from start to completion. The process is broken down into nice bite size chunks so you wont get technological indigestion. In addition we have included an overview of the complete system and information on several NT and Unix utilities which enable you to check that everything is running correctly once you have put it together.

1. Overview of a hotmail system?

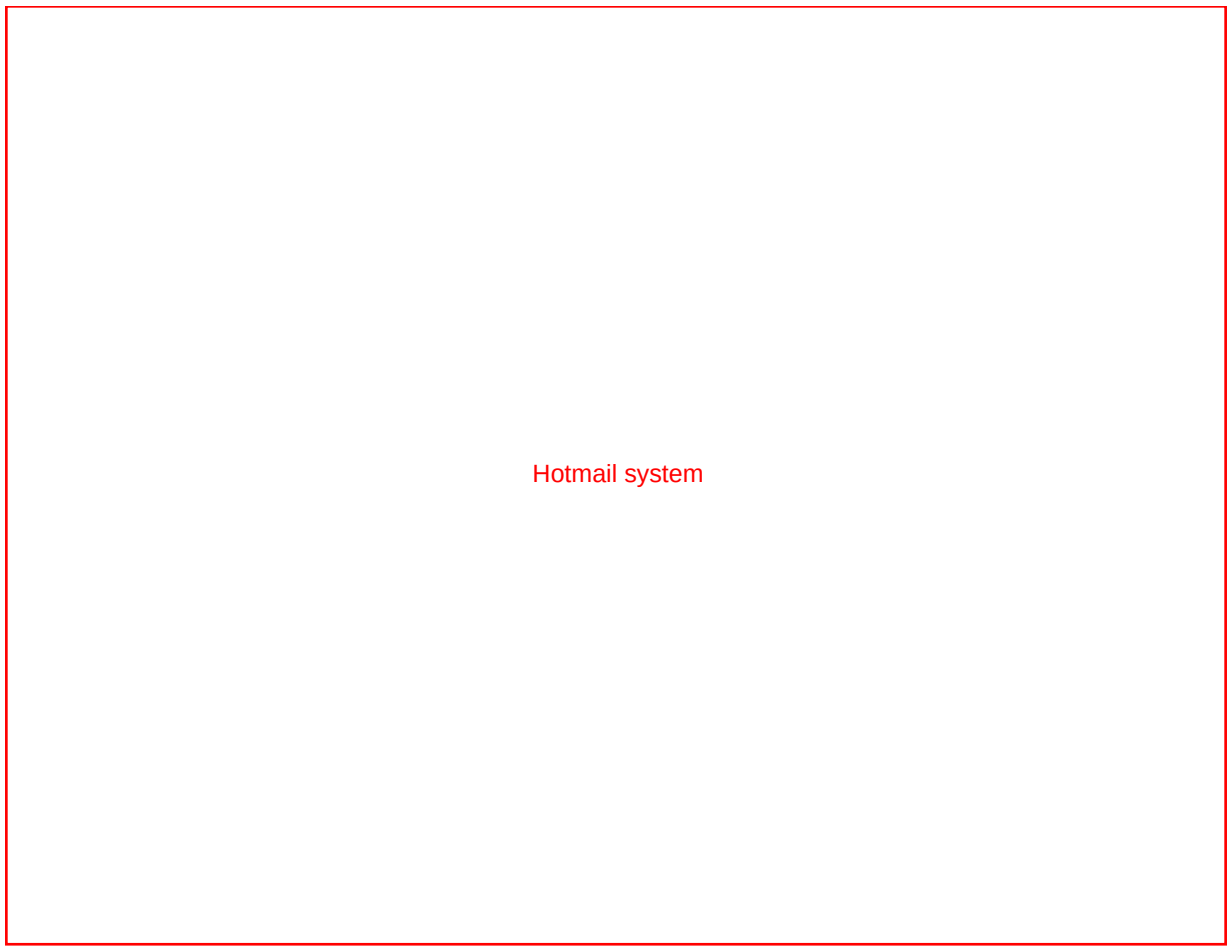
A hotmail system is a way for people to use email via the web.

To provide a hotmail service you need to setup a number of components which will work together. These include standard email services and web interfaces to them. You may also want to allow users to create their own accounts online. You may want to provide a number of domains or virtual domains. You may want to allow other people to administer particular domains. This can all be provided by the Netwin suite of products. You may already have some components of the complete system and just want to add a web interface to them. Provided they obey the relevant standards this should also be possible.

First lets review what your hotmail system needs to include:

- A Web Server of some kind, to run the web interface CGI applications and serve web pages. (CGI's = Netauth, Cwmail, DmailWeb or WebImap)
- A way to receive and send email from other systems, that is an SMTP server. DSMTP is part of our Dmail Mail Server.
- A way to serve email to your users, that is a POP server. DPOP is part of our Dmail Mail Server.
- A way for email from others to find your mail server on the internet, this requires a DNS (Domain Name Server) You need to setup records on an existing DNS or your own DNS. You will need A records, and MX (Mail Exchange) records.
- A way to create email accounts, whether you let users create their own online through a web interface like Netauth or do it yourself offline. You may want to use unix or NT username/passwords or to use an email only database or and LDAP database. Netauth can be configured to create any of these types of accounts.
- A way for users to read and send email, ie a mail client program which operates via the web. Cwmail / DMailWeb or WebImap are three such products. CWMail will normally be used.
- A way to administer your system; Netauth handles user account administration for system administrators, and specific domain administrators, so you can put control in your users hands.

You can create a hotmail system with one domain or multiple domains. A simple system might look like this:



On the left is the outside world (the internet) and a domain name server. On the right inside the large box is everything in your machine or server. Normal access to web mail will be provided via your web server, although direct connections to the email services can also be made from a traditional email client. Inside your server you have a web server, two cgi's CWMail and Netauth, POP and SMTP servers and a database of some type containing usernames and passwords.

Lets just follow through the sequence of events when someone in the outside world sends an email to someone on your hotmail system and then gets a reply:

John on some other system is going to send a message to Sue who has an account on your web mail system:

1. John creates a short message and addresses it to sue@yourplace.com and presses send on his email client.
2. His client software connects to his local SMTP server and gives it the message. It uses a DNS mail lookup to find out where email for yourplace.com should be sent. The DNS server gives his SMTP server the IP address of your machine. The two SMTP servers connect on port 25
3. Your SMTP server first checks it has a user Sue. To do this it checks the user database via and external authentication module. It then accepts the email for Sue and appends it to Sue's drop file.
4. Sue sits at her friends computer opens a browser and connects to http://yourplace.com/scripts/cwmail.exe Again a DNS lookup is used to find your machine. Her web browser connects to your webserver on port 80. Your webserver starts up cwmail.exe as a subprocess gives it the request and waits for it to return a web page in html.
5. CWMail provides a login page requesting username and password. This gets sent to the webserver and then through to the web browser. CWMail.exe then closes down.
6. Sue fills in her username and password and presses login button. Again this is sent to your webserver and then to cwmail.
7. CWMail connects to the POP server to verify account details and see if there is any new mail for her.
8. The POP server checks the user database to make sure password is correct and then checks for a drop file containing email messages which are passed to CWMAIL
9. CWMail passes them back to the web server, then to web browser and Sue selects a message, reads the message and types a

reply. (There are several interactions between the browser, the web server and the cgi to do this)

10. The reply is sent to CWMail which connects to your SMTP server and gives it the reply for John
11. Your SMTP server uses a DNS lookup to find where John is and sends the message to his SMTP server.

This description of a simple interchange between two people probably makes it clear that there are quite a number of components talking to each other on your behalf. To work properly they all need setting up and need to be using the same authentication method and the same usernames etc. So now we need to look at how to install and/or setup each of these components:

Component	Options
DNS	External give them records to insert for you. Run your own, put your own records in.
Webserver	Apache, Peer Web Services, ... Lots of these many good free ones.
POP Server	DPOP...QPOP, other third party poppers
SMTP Server	DSMTP.... QMail, other third party smtp people.
WebMail Interface	CWMail, Dmailweb, WebImap, other third party systems
User/Password database	NT users, Unix users, Netwin's nwauth, LDAP server, SQL database...

There are many possibilities and options in setting up each of these components. To provide a simple concrete example the next section makes a number of assumptions and works through the setup of each component. The following sections provide more detail on each component and an outline of some of the other options available.

Contents

To set up a basic system you will need to follow steps 2, 3, and 4 above they will take you through the setup of Dmail, Cwmail and Netauth creating one email domain. Once completed you can use step 5 to add virtual domains to your mail system.

The very first step in any email system is the setup of DNS entries, you have probably already seen to this if you have a permanent connection to the internet. If you have registered a domain name for your machine say myplace.com then you just have to check that there is also an MX record in the appropriate DNS to ensure that email for anyone@myplace.com will come to your machine. Just ask the person who gave you your domain name if this has or can be done. If you need more detail on setting up these DNS entries then look at section 6 and get this underway before starting on section 2.

There are also two other informational sections 7, and 8 . They briefly describe two utilities which come with most operating systems: telnet and nslookup. These are both very useful when you need to test each section individually to track down a problem.

---

2. Setting up Dmail.

The easiest way to set Dmail up is to concentrate on the host domain first and add virtual mail domains later, the procedure explained below will explain how to set Dmail up for external authentication for use with Netauth, creating a hotmail type system.

Step 1 - Installation.

Go to our dmail download page http://netwinsite.com/dmail/download.htm and download the distribution set for dmail for your system. On NT it is a self extracting archive which will start the installation procedure for you. On Unix you have to uncompress and untar it and then run the setup program.

Install Dmail, following onscreen instructions in the installation program. This will set Dmail up for standard use, no external authentication and one domain.

Step 2 - Add external authentication

To add external authentication to Dmail you need to edit the configuration file, you can do this in two ways using the DMAdmin GUI provided with the windows version of Dmail, or you can edit the dmail.conf file yourself. I am going to explain how to edit the file yourself because this is the only way to configure Dmail on a Unix system and it is very straight forward.

Open the dmail.conf file, it should be found in the system directory on your server (/etc or \WinNT\system32). Find the line "authent_method" and change it's value from "nt_user" or "unix_user" to "external".

Add authent_process setting and set it to "\dmail\nwauth.exe" (Windows) or "/usr/local/dmail/nwauth" (Unix), this is the external authentication process called nwauth and is found in the dmail installation directory.

Add authent_domain setting and set it to "true" this setting instructs dmail to lookup users using domain name and allows you to add virtual domains later on, if you do not set this to true addition of virtual domains later becomes tricky.

While you are here look for the host_domain setting and remember it's value, because you need it in step 4.

Step 3 - Reload configuration

You now need to reload the mail server so that it is running with the new configuration. Go to the system prompt and type "tellpop reload" and "tellsmtp reload", you may need to go to the dmail directory first before these commands will execute.

Setup of dmail should now be complete for use with one mail domain, continue with the setup of Cwmail and/or Netauth before reading the "Adding a virtual domain" section.

---

3. Setting up Cwmail / Dmailweb.

Again just go to our download site http://netwinsite.com/dmailweb/download.htm and download the distribution set for your system. On NT its a self extracting archive which will run the install program for you. On Unix you will have to uncompress and untar it and then run the wmsetup program.

The installation program provided should install Cwmail or Dmailweb without problems. Once it has finished you should be able to connect to http://your.domain/scripts/cwmail.exe or dmailweb.exe OR http://your.domains/cgi-bin/cwmail.cgi or dmailweb.cgi. You should see a login page.

---

4. Setting up Netauth.

Again just go to our download site http://netwinsite.com/netauth/download.htm and download the distribution set for your system. On NT its a self extracting archive which will run the install program for you. On Unix you will have to uncompress and untar it and then run the nasetup program.

The installation program provided should install Netauth without problems. Once finished you should be able to connect to http://your.domain/scripts/netauth.exe OR http://your.domains/cgi-bin/netauth.cgi. You should see the check username page, where users can find out if the username they want is free or already taken.

To test the Netauth try adding a user. To do this enter the desired username "test" into the username field on the first page (the check page), clicking 'check' continues. Assuming this is the first user ever created then the username should be free and you should be given another page (the add page), the name "test" should be entered for you into the username field, all you need to do is enter a password and click 'add'. The user should be added and you should be given a confirmation page. If you receive a page with an error consult the Netauth manual or contact us here at support-netauth@netwinsite.com.

With your new user attempt to log into CWMail. To do this connect to http://your.domain/scripts/cwmail.exe or dmailweb.exe OR http://your.domains/cgi-bin/cwmail.cgi or dmailweb.cgi. You should see a login page. Enter the username "test" and your password and click 'login', the login should proceed with no errors and you should see a first time user page asking for your details. If an error page is returned consult the CWMail manual or contact us here at support-cwmail@netwinsite.com.

---

5. Adding a virtual domain

Once you have Netauth creating users and these users logging into Cwmail you are ready to add a virtual mail domain. To do this you have to add settings to dmail.conf, netauth.ini and cwmail.ini. It may be a good idea to ensure you have backup copies of these files, Netauth and CWMail have already created backups for you in their templates directories. You may want to update these backups when you have everything working correctly.

Remember to modify the .ini files in the web server directory as these are the live copies.

Step 1 - Dmail.conf

"vdomain" is the name of the setting which adds a virtual mail domain, it has this format

vdomain <prefix> <suffix or IP number / name> <domain name> <drop path>

As we are creating virtual domains and are verifying with domain names (authent_domain true) then we don't need to worry about the prefix setting instead the "suffix / IP number" setting is important.

If you have more than one IP number and want to have each domain on a separate IP number then you are creating "IP based virtual domains", otherwise you are creating "suffix based virtual domains".

Here are examples of suffix based vdomain lines.

```
vdomain d2 /domain2 mail.domain2.com \dmail\in
vdomain d2 /dom2 mail.domain2.com /usr/local/dmail/in/2
```

Here are examples of IP based vdomain lines.

```
vdomain d2 1.2.3.4 mail.domain2.com \dmail\in
vdomain d2  2.3.4.5 mail.domain2.com /usr/local/dmail/in/2
```

Of course the actual suffix or IP value will differ depending on your system, and the IP numbers you have available.

Now you are ready to add a vdomain line to dmail.conf, open it and find the "host_domain" setting, I would enter my vdomain lines near this setting, this will make sense to read later on but makes no difference to dmail at all. You can add as many vdomain lines as you like and you are only limited in the case of IP based domains, by the IP numbers you have available. You may not add two vdomains with the same IP number and likewise two vdomains with the same suffix.

So now add a vdomain line, and remember the suffix or IP number for that domain as well as the domain name. If you are using a suffix it is recommended you use a separator character as the first character of the suffix. In my examples I have used a '/' you could use an '@' symbol or another character but the '/' is recommended.

Step 2 - Reload configuration

You now need to reload the mail server so that it is running with the new configuration. Go to the system prompt and type "tellpop reload" and "tellsmtp reload", you may need to go to the dmail directory first before these commands will execute.

Step 3 - Cwmail.ini

After adding you vdomain line(s) you need to tell Cwmail about the domains, to do this you use vhost lines in the cwmail.ini file. The way Cwmail handles virtual mail domains is that it can tell what the URL in the browser is and each URL matches with a different virtual mail domain, as an example.. www.main.com is the URL to the host domain and www.domain2.com is the URL to the first virtual domain you have added then you need a vhost line like this "vhost www.domain2.com".

Vhosts work in sections, Cwmail matches the URL to a vhost line and will then load the settings contained therein until it reads another vhost line or it reads the vend line. Once it has read the vend line it goes back to loading settings normally. So an example ini file may be.

```
[cwmail.ini]
templates \cwmail
pophost 1.2.3.4
smtphost 1.2.3.4

vhost www.domain2.com
pophost 2.3.4.5

vhost www.domain3.com
pophost 3.4.5.6
templates \cwmail\3

vend
nwimg /nwimg
```

In this case the ini file can be loaded in three different ways. If the URL is "www.domain2.com" then the templates pophost and smtphost is loaded then pophost is changed to "2.3.4.5" and then nwimg setting is loaded, if URL is "www.domain3.com" then the same occurs only pophost is "3.4.5.6" and templates is "\cwmail\3". If the URL is anything else then the vhost sections are ignored.

Cwmail gets the URL from the SERVER_NAME environment variable, sometimes this variable is not set to the expected value and if you find you vhost sections are not being loaded you may need to try another environment variable, to tell cwmail to use another environment variable you set the vhost_match setting to the name of this variable, like "HTTP_HOST" which can also return the URL host.

You can put any settings you like into a vhost section and these variables will either replace or add to previous settings, in the case of the templates setting the new value replaces the old one, but in the case of the body_add setting the new value is added to the list of values.

Now you will need to add a vhost section to match the new virtual domain added above. If using suffix based domains then the vhost section will need to contain a suffix value, the same value you used in the vdomain line above. If using IP based domains then a pophost, and smtphost settings with the new IP number are required. And remember to include a domain setting in the vhost section stating the new virtual domain name, this should match the domain name from the vdomain line.

Step 4 - Netauth.ini

Netauth also uses vhost sections to set up virtual domains, it does this in the same way as cwmail and also uses the vhost_match setting if required.

You will need to add a vhost section to match the new virtual domain. If using suffix based virtual domains, you will need a suffix setting, unlike cwmail early versions of Netauth Netauth (version 3.0e and earlier) use two settings for the suffix. "suffix" and "suffix_seperator", the suffix setting is the same as the value used in the vdomain line minus the separator character. And suffix_seperator is the separator character which defaults to / so you do not need to set it unless you have used another character in the vdomain line. Later versions (Netauth 4.0+) suffix setting will be the same as Cwmail's suffix setting. If using IP based virtual domains you will need to specify a new pophost setting with the new IP number, this is identical to cwmail. Remember to include a domain setting in the vhost section stating the new virtual domain name, this should match the domain name from the vdomain line.

---

6. DNS Entries, MX entries and A records

DNS management requires two name servers a primary and a secondary, this is because the InterNIC will not grant you a domain name unless there are at least two DNS servers on the Internet with information about that domain. Another good reason is if you only have one and it goes down then users will be cut of from the internet. This gives you three options for DNS management, they are...

1. Use your ISP's primary and secondary name servers.
2. Use your own primary and secondary name server.
3. Use one of your ISP's name servers and manage the other yourself.

Option 1

To do this first you must inform your ISP that it is providing both primary and secondary DNS services for you, if they are unable or do not wish to do this then you will have to use option 2. Next step is to inform the ISP of the DNS records you want to publish, to allow interaction with your network. In addition you will need MX records if you want to receive mail at that domain and A records for your ftp and www server(s). You will need to inform your ISP of these addresses also.

Option 2

There are several reasons for running your own DNS server, they are

1. You ISP will not allow you access to theirs, or does not have one.
2. You own a part of the internet (namespace).
3. If you are running IP network-based applications inside your network that require users to connect to internal machines by name, and you don't want to advertise the names and/or addresses of these machines to possible hackers.
4. You want full control, you own DNS server means you can make immediate changes, you do not have to wait while your ISP changes things for you.

You will need to purchase DNS software, as most servers are run on Unix machines the most popular software is BIND (Berkeley Internet Name Domain), here you will find software and training in DNS management.

Option 3

This option has many of the same principles as option 2. And there are two more choices here..

1. You manage the primary name server.

2. You manage the secondary name server.

If you choose to manage the primary name server yourself, keep in mind that you'll have to maintain the DNS records.

If you choose to have a secondary name server onsite, then your ISP will still do all of the work, and your server will simply download the data about your domain from the primary server periodically.

DNS Records

Two types of DNS record are MX Records, for mail delivery and A Records, used to resolve computer names to IP numbers.

MX Records

MX Records or Mail Exchange records are used to allow mail delivery, they are in this format

<domain_name> <machine_with_mail_server> <preference>

i.e. An MX record might contain
netwinsite.com netwin.co.nz 0;
netwinsite.com mail.netwin.co.nz 1;
netwinsite.com mail3.netwin.co.nz 2;
In this example mail delivery would be to netwin.co.nz first, because it has the lowest preference value, if this fails then to mail.netwin.co.nz and finally to mail3.netwin.co.nz.

A Records

A records associate computer names with their actual IP address, they are in this form

<computer_name> <address>

i.e. An A record might be
netwin.co.nz 1.2.3.4
In this example the DNS server will locate netwin.co.nz and use the IP 1.2.3.4 to locate the machine.

You require A records for all the machines mentioned in MX records, all machines where you wish to use ftp (File Transfer Protocol) and all machines running a www (World Wide Web) server.

---

7. Using 'Telnet'

Telnet is a utility provided with both NT and Unix systems. It allows you to connect directly to IP addresses on any port. With this utility we can check...

- SMTP port.
- POP port.

Basic operation of telnet

Telnet differs a little from the windows system to Unix, there are two sections describing its use, one for windows and one for Unix.

Windows - Telnet

To execute telnet type "telnet" at the command prompt or click 'Start', 'run' and type "telnet". You should be presented with a window titled "Telnet -(none)" this is Telnet. To connect to a machine simply open the 'Connect' menu and choose 'Remote System' you will be presented another window containing three fields, 'Host name', 'Port' and 'Term type'. The 'Host name' is the name or IP number of the machine you want to connect to. The 'Port' field is the port number you wish to connect to on that machine. Ports include '23' telnet port, '25' smtp port, and '110' pop port, and there are several others. Once connected you can type data directly to the port and receive replies, this provides an excellent way of ensuring things are functioning as they are supposed to.

Unix - Telnet

To execute telnet type "telnet" at the command prompt, then you will be presented with a 'telnet>' prompt type "?" and it will display it's list of available commands. To connect to a machine type "open <machine> <port>", where <machine> is the IP number or name of the machine, and <port> is the port you wish to connect on. Ports include '23' telnet port, '25' smtp port, and '110' pop port, and there are several others. Once connected you can type data directly to the port and receive replies, this provides an excellent way of ensuring things are functioning as they are supposed to.

In order to check the various ports on your system you have to know a little about how each port works, here is a brief explanation of how to *talk* to the pop port.

POP port

The pop port is used to check / receive / delete mail from the mail server, it is port number 110, there are several commands that can be used while logged into the pop server, but first you must log into pop.
To do this type "user <username>", <enter>, "pass <password>", <enter>. Of course this requires an existing user and correct password. Once logged in you can use these commands.

Command	Example	What it does?
list	list	The list command lists a user's email, it lists them with a reference number which can be used to retrieve / delete the message. This reference number only lasts until you log out and a different message may have the reference number next time.
uidl	uidl	This uidl commands lists messages by their UID (Unique ID) number, this is useful because this number unlike the reference number above never changes, once allocated until the message is deleted.
retr	retr 1	The retr command retrieves the contents of message with the given reference number, in this example message 1.
dele	dele 1	The dele command deletes the message with the given reference number, in this example message 1.
top	top 1 10	The top command retrieves the top x line of the message with the given reference number, in this example top 10 lines of message 1.
quit	quit	This commands disconnects you from the pop server.

---

8. Using 'Nslookup'

Nslookup is a utility that looks up DNS entries, it is the easiest way to check whether your DNS entries are correct. This utility differs very little between operating systems so one set of instructions should suffice.

To execute nslookup, type "nslookup" and press enter.
You should see something like this...

```
Default Server : ns-100Mb.webpros.com
Address : 206.127.192.1

>
```

There is a list of help on nslookup commands, to get this list type "?" and press enter.
Otherwise you can simply type the domain name into the prompt and press enter to search for it.

```
Default Server:   ns-100Mb.webpros.com
Address:   206.127.192.1

>netwinsite.com
Server:   ns-100Mb.webpros.com
Address:   206.127.192.1

Name:   netwinsite.com
Address:   207.230.97.10

>
```

You can also search for MX records...

```
Default Server:   ns-100Mb.webpros.com
Address:   206.127.192.1

>set type=MX
> netwinsite.com
Server: ns-100Mb.webpros.com
Address: 206.127.192.1

netwinsite.com preference = 10, mail exchanger = eagle.webpros.com
netwinsite.com preference = 20, mail exchanger = falcon.webpros.com
netwinsite.com nameserver = ns.webpros.com
netwinsite.com nameserver = ns.professionals.com
eagle.webpros.com internet address = 206.127.192.10
falcon.webpros.com internet address = 206.127.192.2
ns.webpros.com internet address = 206.127.192.1
ns.professionals.com internet address = 207.230.127.126
>
```

Setting Up a Web Based Email system with Auto Account Creation, e.g. a "HotMail" Type System

NB: This page is part of the DMail System Administrator Manual

To set up a Web Based Email system where users can add themselves without intervention from the system administrator (like 'HotMail') you will need to set up:

1. An email server
2. A web to email interface for the users to read their mail.
3. Then you need to set up some way for the users to add themselves as users to the email system (and you also may want this to include options for users to change password or to set up forwarding rules, autoresponders, mailing lists etc.).

Our Solution (presented on this page):

1. The Email Server: We produce the DMail email server which is a good product to use as the email server for such a setup.

2. The Web-EMail Interface: We produce CWMail (and DMailWeb) as an excellent email to web interface solution.

3. The Web EMail Account Creator: We now produce NetAuth which allows users to add themselves to the email server system (and much more) from a web page.

Each part of this system that we produce is an individual fully featured product.

The rest of this page has details on how each of these three parts relate to each other to setup the Auto Account Creation system and links to relevant information in each one's section of our site.

Table of Contents for this page:

These links may also be useful:

- **Step by Step Installation Guide**
- What is an Auto Account Creation system?

- FAQ - Web Based Email Systems with Auto Account Creation

Special Note on Cost:

This is not an expensive system, approximately
US$700
for a large (unlimited user) system!

You can register DMail and CWMail (or DMailWeb) for a BUNDLE price. NetAuth comes free with all DMail licenses.

See our prices page for details.

The Email Server - DMail

You will find more than enough 'trumpet blowing' on the
DMail Home Page

The Web Email Interface - CWMail (or DMailWeb)

We produce two options for the Web Email Interface part of a Web Email system, CWMail or DMailWeb.

Basically CWMail is a more fully featured version of DMailWeb and each has its niche area:

- CWMail allows the user to have a fully featured email client, so it suits users who will mostly access their mail from the web.
- DMailWeb on the other hand simply has the necessary email client features, and suits users who want occasional access to their email from the web.

Both are CGIs (commonly known as 'scripts') that run on your web server.

They come with a set of template web pages which are fully customisable so that you can make this 'web based email client' look like the rest of your site.

See
CWMail/DMailWeb Home Page

The Web EMail Account Creator: - NetAuth

The best option for the Web based email account creation part of the system is NetAuth.

NetAuth is a web based email account creation/administration tool which provides:

1. email account creation and administration for users
2. email domain adminstration tools for domain administrators
3. email system administration tools for the sys admin

Some examples of its capabilities are,

- adding users
- changing passwords
- setting up aliases
- creating mailing lists
- etc.

It is a new beta product (free with the DMail 'Unlimited User License') , with its own web page at, http://www.netwinsite.com/netauth

NetAuth is a CGI that runs on the web server (in the same way that CWMail and DMailWeb do) and has a set of web pages (templates) which you can customise.

Note: NetAuth is fully customisable as per CWMail/DMailWeb - it is easy to make NetAuth look as good as the rest of your site :-)

The other option for Web based email account creation is WAdduser.

Note: WAdduser was an early limited form of NetAuth which will slowly be phased out. WAdduser can be used to create and optionally delete mail accounts from the web. For more details see,
Setting Up WAdduser

So how does the NetAuth CGI add users to the email system?

NetAuth 'talks' to an external authentication module which in turn adds the users to the user database.

Up until now NetAuth required that you use NWAuth (free with DMail) as your user database. NWAuth is Netwin's own simple but very efficient user database program. It comes in the DMail distribution set, along with its source so that you can modify it.

Version 2.0 of NetAuth works with external authentication modules other than NWAuth - e.g. our LDAPauth or a module of your own design or modification (e.g. the ODBC authenticator). (In the near future it will also work with system password files, i.e. Unix style /etc/passwd and NT's system user database.)

We still recommend that you use NWAuth if you do not have any limitations on which authentication option you choose. For details of its performance see,
Performance Statistics

I want to make use of the Free Trial period to try this out.
So how do I set this up?

For those who want simple 'do this' type instructions we now have the,
Step by Step Installation Guide

For those who need to know more about how the system works, we recommend the following sequence:

1. Download and install DMail. Set it up for just ONE domain (add extra 'virtual' domains later)
2. Set up External Authentication (nwauth) with 'user@domain' type authentication:
   for example, set/edit the following settings in dmail.conf (typically c:\winnt\system32\dmail.conf or /etc/dmail.conf),

   authent_method external
   authent_process c:\dmail\nwauth.exe
   authent_domain true

   Notes:
   - there should be a host_domain setting in dmail.conf for your domain, e.g., host_domain mydomain.com
   - you need to RE-START DMSTP and DPOP after changing the authentication process.

3. Add a couple of test users:

   You can do this either using the 'Users' section of the windows GUI admin tool, DMAdmin, or you can run nwauth from a command line, e.g. to add bob and fred to the domain, mydomain.com enter,

   c:\dmail\nwauth (runs nwauth)
   set bob@mydomain.com pass1
   set fred@mydomain.com pass2
   quit (to close nwauth)

   Now check that you have successfully added the two users by opening in a text editor the database file, nwauth.add (or nwauth.txt). You should see two lines (with encrypted passwords) like,
   bob@mydomain.com:eqYFAFM:
   fred@mydomain.com:byIMPKK:

   At this point you should be able to send and receive email with a normal, non-web-based, email client.
4. Download and install CWMail (or DMailWeb)

   Now you should be able to send and receive email using the web interface.
5. Download and install NetAuth

   Now you should have a complete system. You should be able to add users from the web.

Each of the products has its own information on installation, which you should read (click on the links above). Our products are rich in features so don't get bogged down in the details :-)

If you have any problems or this does not cover your specific situation then please contact
DMail support

or for CWMail/DMailWeb specific queries you can contact,
CWMail support

similarly for NetAuth specific queries you can contact
NetAuth support

- any of us will be happy to help :-)

What is a Auto Account Creation system?

A very popular feature of the DMail suite of products is the ability to setup a Web Based Email System with Auto Account Creation, for example a 'HotMail' type system.

Users Adding Themselves to Your EMail System:

Basically such a system is where users can go along to your web site and create themselves a mail account. Often they are asked to provide details about themselves.

Users can then access this mail account via the web, so that they can send and read mail from anywhere in the world. All they need is a computer with a web browser and an internet connection.

Find out how to set up an Web Based Email system

LDAPAuth External Authentication Module

The LDAPAuth module is for use with Netwin's DMail and DNews products and an LDAP server. Several of Netwin's products allow the use of a separately compiled authentication module which obey a simple command protocol (see External Authentication). Netwin provides several such authentication modules and you can also build your own. LDAPAUTH is one of the modules Netwin provides and enables you to authenticate against an LDAP server rather than standard NT or Unix password files.

Contents:

1. Use with DMail

Directory severs contain information about objects. One sort of object is a person and the various bits of information stored in the directory server about that object are called attributes. The attributes are stored will vary from one server to another. The key ones for use with ldapauth is the mail attribute which has the form user@domain this is what ldapauth refers to as the username. The distinguished name or dn is generally of no interest to ldapauth. The @domain part of the username is optional and will default to a setting in the ini file.

2. Use with DNews

Same as with dmail, except that the entry in dnews.conf should be:
auth_spawn d:\dnews\ldapauth.exe
instead of the dmail settings
authent_method External
authent_process c:\dmail\ldapauth.exe

See http://netwinsite.com/dnews/access.htm#pass_ldap for the few dnews specific instructions but in general follow the dmail installation instructions.

3. LDAPAUTH Command Set:

The ldapauth module receives commands on stdin and sends replies on stdout. The command set is given below:

NB: Ldapauth does not take any command line arguments.

NB: LDAPAuth's command set is defined by the Netwin External Authentication Proctocol.

Command	Function
lookup username	Retrieve drop file location and other information about an already authenticated user
check username password	Check the a username and password against LDAP directory
search username	Search for matching users
set username password option=" value"	Create a new user entry in the LDAP directory
quit	close down ldapauth
exit	close down ldapauth
verbose	Toggle verbose mode. Verbose mode is useful for testing when running ldapauth at the command prompt. It outputs additional information in response to each command
version	Print the version number

Replies consist of:

+OK message

-ERR message

+DATA message

Details of the response and additional options within each command are given in the following sections:

3.1 Check Command

Check the a username and password against LDAP directory. The username may be either user@domain or just user. The LDAP directory will be searched for entries with the attribute mail = user@domain. If

the supplied username has no @domain section then this will be taken from the ini file setting pop_domain. Once a matching entry is found then ldapauth will try and bind to that entry using the supplied password. The fromip parameter is currently ignored by this module, it is used in other modules to specify different virtual domains. If the bind operation is successful the +OK is returned if not -ERR is returned.

If the bind was successful then the LDAP entry is also checked for drop file, uid and forward attributes. If these are found then they are also returned. If no drop path attribute is found the word config is returned in its place. If no uid attribute is found 0 is returned. Forward addresses, if found, are returned at the end of the +OK line. The attributes to be used for drop_file, uid and forward are specified by the ini settings ldap_drop_file, ldap_mail_uid and ldap_dmail_forward. The default values for these are: drop_file, mail_uid and mailForwardingAddress.

Input:

> check username@domain password fromip
> check username password fromip

Returns:

> +OK username path uid forward
> -ERR reason

3.2 Exit Command

Shuts down the ldapauth module.

3.3 Help Command

List the available commands using the format
+DATA information
+DATA information
+OK

3.4 Lookup Command

Retrieves drop file location uid and forward information about an already authenticated user. The format etc. is similar to that used in the check command but no password is required. This is used for example by DPOP when it needs to check the location of a drop file without first authenticating a user.

Input:

> lookup user@domain fromip
> lookup user fromip

Returns:

+OK username path uid forward
-ERR reason

3.5 Search Command

Search for matching users. The supplied username may be of the form user@domain or just user. The LDAP directory will be searched for entries with the attribute mail matching user*@domain. If the supplied username has no @domain section then this will be taken from the ini file setting pop_domain. To check for users from any domain use: search user@* One line is returned for each matching entry with a +OK or -ERR line at the end.

Input:

search user
search r
search *smith
search Ralph@*

Returns:

+DATA dn=(cn=Test0, o=netwin), mail=test0@161.29.2.44
+DATA dn=(cn=Test1, o=netwin), mail=test1@161.29.2.44, path=/spool/mail
+OK Search Complete

-ERR No matching entries

3.6 Set Command

This command has several uses depending on the format used. It can:

set newuser password	add a new user and password to the LDAP server
set olduser newpassword	change the password of an existing user
set olduser (NULL) age="45"	add new attributes for an existing user
set olduser (NULL) age="46"	modify the values of attributes for an existing user

When it is adding a new user it will use the ini setting ldap_objectclass setting to specify what the object to be added is. The default is a person. They will have a dn: newuser as well as an attribute mail: newuser@domain where @domain was either part of the specified username or taken from the ini setting pop_domain.

When adding new attributes they are specified in the form name="value" the quotation characters are

required and multiple attribute name="values" can be specified separated by spaces.

NB: You MUST ensure that any attributes that you supply to the set command exist in your objectclass, otherwise you will get an error like,
-ERR ld_set: Cant modify entry for cn=fred: Object class violation

If the password is not to be changed or rewritten to the LDAP directory then the password is specified as the string (NULL)

The modify attribute value has the same format as the add new attribute. Note that the new value replaces any existing values.

Input:

> set username password|(NULL) [var="value" ...]
> set user password
> set user (null) age="45" sex="male"

Response:

> +OK Database modified for user (mail=ralph@161.29.2.44)
> -ERR ld_set: Cant modify entry for cn=Ralph, o=netwin: Protocol error

---

3.7 Quit Command

Shutdown ldapauth module. Identical to exit command.

3.8 Verbose Command

Toggles the verbose mode. In verbose mode it is assumed the module is being run at a command prompt and additional information can be printed. For example lookup and check will return the full information of all attributes for the selected entry.

---

4. Ini Settings for LDAPAUTH

Netwin supplies several external authentication modules:

- nwauth
- ldap1

These are all supplied with source so they can easily be tailored to your particular use. The LDAPAUTH module allows you to use an LDAP server to authenticate against in addition to storing such things as the location of the users drop file in the LDAP database. It makes use of an ini file which must be called ldapauth.ini an must be stored with the ldapauth executable. The settings which can be used within the

ini file are given below:

Setting	Example	Default	Required	Function
ldap_port	3890	389	no	The TCPIP port to connect to the LDAP server on
log_path	c:/logs	the location of the ldapauth executable	no	The directory to store log files in
max_log_size	10000	100000	no	The size at which log files are rotated. Logs are numbered 1,2,3,4
log_level	debug	info	no	Controls the amount of information logged during use. One of error, info, debug
ldap_host	whatsit.co.nz	localhost	required	The IP name of the host to connect to
ldap_mail_uid	dmail_uid	mail_uid	no	The LDAP attribute which will be used to store the uid DMail should use for accessing the users drop file. Optional
ldap_mgr_dn	cn=DMAIL Manager	cn=Directory Manager	required	The LDAP manager distinguished name to bind with.
ldap_mgr_pw	secret	blank	required	The password for the ldap_mgr_dn entry

ldap_search_base	o=whatsitinc	blank	required	The LDAP search base to use for all interactions with the LDAP server
pop_domain	whatsit.co.nz	blank	required	The domain which will be appended to any usernames not containing @domain
ldap_drop_file	dmail_drop_file	drop_file	no	Name of LDAP attribute which will be used for storing the full name and path of the users drop file. Optional
ldap_objectclass	android	person	no	When adding new users with the set command the specified objectclass values will be added to the new entry, NB: you must enter something like, set username pass a="x" b="y" c="z" , where the fields a, b and c MUST be in the specified objectclass, otherwise you will get an objectclass violation error message.

ldap_dmail_forward	dmail_forward	mailForwardingAddress	no	Name of LDAP attribute which will be used to store DMail forwarding addresses
log_name	ldapauthlog	ldapauth	no	Base of log file name. Note suffix n.log will be appended so default is ldapauth1.log

---

5. NT Installation with DMail:

You need to perform the following steps:

1. Download the distribution set
2. Unpack the distribution set
3. Copy ldapauth.exe and ldapauth.ini to a directory where DMail can use it.
4. Edit ldapauth.ini to meet your requirements.
5. Update dmail.conf to tell dmail to use ldapauth for authentication.
6. Restart DMail.
7. Test it.

These steps are described in detail below:

1.

NB Windows users: With versions 2.8h and above of DMail you will find the ldapauth files in your DMail distribution set (\dmtemp). They should have been copied into your DMail directory by the DMSetup utility, so you should check step 3 below and then jump to step 4.

Download the distribution set
For the latest versions see, Utilities Download Page
General command line FTP instructions:
ftp ftp.netwinsite.com
(log in with username 'anonymous', use your email address as a password)
cd pub/dmail
hash
binary
get ldp10c.exe

2. Unpack the distribution set
ldp10a

3. Copy files (ldapauth.exe etc.) in temporary unpack directory /ldtemp to a directory where DMail can use them.

NB: You MUST remember to copy the ldap dll

```
cd \ldtemp
copy * \dmail
```

You should find the following files,
ldapauth (binary)
nsldap32v11.dll (copy this to dmail directory or c:\winnt\system32\)
ldapauth.ini (sample ini file for you to edit)
ldap.htm (a copy of this page)
authprot.htm (outlines our External Authentication Protocol)

4. Edit ldapauth.ini to meet your requirements.
notepad \dmail\ldapauth.ini
! See other sections of this page for configuration information !
Note ldapauth.ini will normally contain the manager password for your ldap server so it should be suitably protected but ldapauth must be able to read it. LDAPAuth is spawned by DPOP which is spawned by the DWatch service which is normally run as the 'system' account.

5. Update dmail.conf to tell dmail to use ldapauth for authentication.
notepad \winnt\system32\dmail.conf
add/modify the following lines:

```
authent_method External
authent_process c:\dmail\ldapauth.exe
```

6. Restart DMail.
Use dmadmin utility for this. Do a stop all and then a start all.
or see Restart information

7. Test it.
Try a search or add user from dmadmin or just connecting from an email client with a username/password which exists on the ldap server. If these tests fail it may help to run the ldapauth.exe from the command line then use comands; help and verbose etc. to see what the problem is.

---

6. Unix Installation with DMail:

You need to perform the following steps

1. Download the distribution set
2. Unpack the distribution set
3. Copy ldapauth.exe and ldapauth.ini to a directory where DMail can use it.
4. Edit ldapauth.ini to meet your requirements.
5. Update dmail.conf to tell dmail to use ldapauth for authentication.
6. Restart DMail.
7. Test it.

These steps are described in detail below:

1.

NB Linux Libc6 users: With versions 2.8h and above of DMail you will find the ldapauth files in your DMail distribution set (/dmtemp). They should have been copied into your DMail directory by the DMSetup utility, so you should check step 3 below and then jump to step 4.

Download the distribution set
For the latest versions see, Utilities Download Page
General command line FTP instructions:

```
ftp ftp.netwinsite.com
(log in with username 'anonymous', use your email address as a password)
cd pub/dmail
binary
hash
get ldp10c.exe
```

2. Unpack the distribution set by entering at the command prompt,

```
ldp10c
```

3. Copy files in temporary unpack directory /ldtemp to a directory where DMail can use them, i.e. to the dsmtp_path.

```
cd /ldtemp
cp * /usr/local/dmail
```

You should find the following files,

```
ldapauth (binary)
ldapauth.ini (sample ini file for you to edit)
ldap.htm (a copy of this page)
authprot.htm (outlines our External Authentication Protocol)
```

4. Edit ldapauth.ini to meet your requirements.

```
vi /usr/local/dmail/ldapauth.ini
```

! See other sections of this page for configuration information !
Note ldapauth.ini will normally contain the manager password your ldap server so it should be suitably protected but ldapauth must be able to read it. NB: LDAPAuth will normally be able to read a protected file as it runs as root, when spawned by DPOP or DSMTP.

5. Update /etc/dmail.conf to tell dmail to use ldapauth for authentication.
vi /etc/dmail.conf
add/modify the following lines:
authent_method External
authent_process /usr/local/dmail/ldapauth

6. Restart DMail.
Use dmadmin utility for this or
tellsmtp shutdown
tellpop shutdown
/usr/local/dmail/dm_start.sh
/usr/local/dmail/dpop_start.sh

7. Test it.
Try a search or add user from dmadmin or just connecting from an email client with a username/password which exists on the ldap server. If these tests fail it may help to run the ldapauth.exe from the command line then use comands; help and verbose etc. to see what the problem is.

---

7. Building for other platforms:

It is now our policy to provide a binary form of ldapauth for as many platforms as possible. (For linux and NT you will find ldapauth binaries in the DMail distribution set).So before building LDAPAuth for yourself, you should check that our utilities page does not have a precompiled executable for you. And if it doesn't you can of course email us and we will try to build it for you.

We will still provide the source, downloadable from our utilities page.

Below are instructions for compiling ldapauth yourself.

1. Download the ldapauth source from our utilities page,
http://www.netwinsite.com/dmail/utils.htm#ldapauth
,or directly from our ftp site, login as user 'anonymous',
ftp://ftp.netwinsite.com/pub/dmail/ldp10c_source.tar.Z
(or a similar file name)

2. After downloading the file you would do this:
uncompress ldp10c_source.tar.Z

tar -xvf lpd10c_source.tar
cd buildxx (e.g. build28)
cp config.PLATFORM config.i
(where PLATFORM is your platform, e.g. solaris)

3. If you don't have the sdk installed...

3.a) You can install the LDAP C libraries sdk from:
http://www.openldap.org/
or (Netscape) http://www.mozilla.org/directory/

and then edit the SPECLIBS line of config.i so that you include the correct libraries for the SDK that you installed, e.g. libldap and liblber, e.g.,
SPECLIBS = -I../ldap/include -L../ldap/libraries -lldap -llber

OR

3.b) Our ldapauth source contains the openldap sdk, so do the following in the buildxx directory,

*** you may need to edit config.i at this point and uncomment SPECLIBS line ***

uncompress ldapopen-src.tar.Z
tar -xvf ldapopen-src.tar
cd ldap
./configure
make depend
make

(that should make the libraries, see INSTALL for OPENLDAP instructions on installing SDK)

cd .. (back up to buildxx directory)

4. Now LDAPAuth should build correctly:
cd dpop
make ldapauth

If not check that you are including the SDK include directory and providing the location of the two SDK libraries, libldap and liblber, which should be on the SPECLIBS line in the config.i you are using, e.g. something like,
SPECLIBS = -I../ldap/include - L../ldap/libraries -lldap -llber

(ensure that the line in your config.i is not commented out with a pound or hash, #, symbol).

5. Then use the resulting ldapauth binary with dmail or dnews.

For DMail,
copy ldapauth to your dmail directory,
cp ldapauth /usr/local/dmail

copy other files to dmail directory,
cp ../misc/ldap* /usr/local/dmail

cd /usr/local/dmail

edit ldapauth.ini (see the config settings on this page for details)
vi ldapauth.ini

6.Test ldapauth at the command line,
./ldapauth
lookup username
+ok username ...
quit

---

8. Downloading a recommended LDAP server

We recommend the OpenLdap server available from,
http://www.openldap.org/

We also test LDAPAuth against Netscape's LDAP server (a number of the defaults relect this) and the University of Michigan server.

SDKs for LDAP are also available from,
http://www.openldap.org/
OR (Netscape) http://www.mozilla.org/directory/

POPFetch - The DSMTP 'side salad'

Under Construction :-)

Note POPFetch is a new DMail add on that is still in beta form - it should not be used unless you are instructed to do so by DMail support!

Introduction

PopFetch is a "side salad" for your SMTP server. It runs beside your SMTP server and works with it to provide email for an intranet without a constant internet connection.

On Windows platforms it can automatically dialup to your ISP (RAS dial). This provides economical use of your internet connection.

PopFetch is a perfect solution for email at your remote offices.

So what does it do ?

PopFetch periodically connects to a POP server and collects all mail for a specific email account. It then feeds that mail to a local SMTP server so that it is ready for collection by email clients on the local network.

When PopFetch feeds the messages to the SMTP server it addresses them to the same recipient

PopFetch can be set to change the domain name of the recipient address(es) before it passes them on to the SMTP server so that the SMTP server delivers them to local users, e.g. if PopFetch collects messages from a POP server at netwin.co.nz ,

say for john@netwin.co.nz,

and PopFetch is set to re-address for say branch_office.netwin.co.nz, then it will re-address the message to john as,

john@branch_office.netwin.co.nz

In conjunction with Netwin's DSMTP, SMTP server . . .

PopFetch works in conjunction with DSMTP (Netwin's SMTP server) so that when it starts a dial up networking connection, DSMTP can deliver all mail waiting to be sent to non-local addresses.

What happens when a message cannot be delivered?

POPFetch always copies the current message that it is retrieving to file (msg.dat), it then tries to send the message immediately on to the SMTP server. If for any reason this delivery fails, POPFetch stops and closes all TCPIP connections and waits for the next dial up time, at which point it will try to send the unsuccessful message again.

If the message cannot be delivered, because the information stored in msg.dat is corrupt, or the message cannot be sent to any users - even the default address to say that a user cannot be reached, then POPFetch will write the message to a 'duds' file, duds.dat. There is a safety net setting, max_duds with a default of 10, such that if POPFetch detects that it has delivered the max_duds number of messages in a row to duds.dat then it will shut itself down (stop the popfetch service on NT).

Notes:

- If a message fails to be delivered to the SMTP server, POPFetch stops and closes all TCPIP connections and waits for the next dial up time, at which point it will try to send the unsuccessful message again.
- Every time the ras_timer period is reached, popfetch will try to retrieve mail for all accounts specified by a pf_user line.
- If a message cannot be delivered, for example if the recipient does not exist or the SMTP server rejects the recipient address, then the fallback address will receive the message instead. POPFetch adds some informational lines to the message body stating which recipients have received the message and which have not.
- log_path \dmail\log (\getall.log)
- work_dir defaults to no path, then log_path def to work dir
- Ras timer sets period of dial rather than interval, by default (isinterval=FALSE)
- POPFetch knows that the next line of an SMTP response is part of the same comment when the fourth character of the response is a dash, '-'.
- reads headers that go over multiple lines. The only header where this matters for POPFetch is the header set by pf_header
- if the pf_header setting cannot be found in the message to be sent out, POPFetch looks for the 'X-Rcpt-To: ' header, followed by the 'To: ' header, then if even that header cannot be found it will send the message to the default address, if that does not exist then the message will be appended to the duds file, duds.dat
- -DFLT_WORK_PATH does not exist - i.e. use exe directory
- minimum ras_timer setting of 5 minutes if doing RAS dial, i.e. if you have a ras_entry or ras_number setting, otherwise there is no mimimum setting.

- The ras dial up is considered to have succeeded if another program is controlling the dial session - there is a pf_wait setting to set a 'try-again' time to counter the fact that the other process might still be doing the dialup. This wait period is only used if POPFetch cannot reach the servers when it wakes up.
- If a message fails to be delivered to multiple recipients then POPFetch will only send one message to the pf_dflt_address, with a paragraph added to the top of the message specifying which recipients did not receive the message.

Features:

- Can connect to multiple POP servers and/or multiple accounts.
- Can be set to do RAS dial up to the internet on Windows platforms.
- Can alter the domain in the destination address.
- Can be set to read destination address from any header,
  e.g. To:, X-Recip-To: etc.
- Can be run as service on Windows NT
- specifiable caretaker address for failed messages to be delivered to.
- Built in "safety net" shutdown in case of misconfiguration
- Detect when a RAS dial connection is already open, including a configurable 40 second delay if the POP connection fails due to another program still being in the dialling process.

Also in conjunciton with DSMTP, Netwin's SMTP server, POPFetch can . . .

- Make use of DSMTP's X-Recip-To: header, created from the RCPT TO: line of the SMTP envelope, to get accurate message delivery.
- Use DSMTP's RAS dial ini settings (set in dmail.conf)
- DSMTP automatically sends the Extended SMTP, ETRN command, when the dial up connection is made, to initiate the remote server to send any mail waiting for the local domains, including virtual domains.

POPFetch Settings

POPFetch reads DMail's configuration file, dmail.conf, to find its settings. E.g. \winnt\system32\dmail.conf on Windows platforms and /etc/dmail.conf on Unix based platforms.

On all platforms, no matter whether you want RAS dialup or not, POPFetch uses the setting ras_timer x to decide when it should next check for new mail. So this is the first setting you should set. Settings

specifically for POPFetch all begin 'pf_' in dmail.conf.

On windows NT it can also use the other DSMTP RAS Dial settings, to do RAS Dialup before each check so that it can reach the POP server from which it is to download mail.

The settings relevant to POPFetch are:

POPFetch will create a log file, popfetch.log, in the directory and at the level set by the DMail settings, log_path and log_level. See Other DMail Settings That POPFetch Uses below.

RAS Settings:

- ras_timer period for checks
- ras_number phone number to dial
- ras_entry name of RAS entry to be used.
- ras_username username for dial up connection
- ras_password password for dial up connection
- ras_domain domain to authenticate with on dial up (leave blank)
- ras_smtpip IP address of SMTP server DSMTP should talk to (not used by POPFetch)
- ras_timeout timeout value for RAS connection(not used by POPFetch)

POPFetch specific settings:

- pf_user
- pf_header
- pf_isinterval
- pf_domain
- pf_header
- pf_max_duds
- pf_dflt_address
- pf_retry_delay

Other DMail settings that POPFetch uses:

- log_path
- max_log_size
- log_level
- smtp_timeout
- pop_timeout
- work_path

Details of POPFetch specific settings:

- pf_user <username> <password> <pop_ip_address>

  You can have multiple pf_user lines. Each line sets an email POP account that POPFetch should check for mail. As such the pf_user setting takes parameters of the username, password and IP address for the POP server where the mail account exists.

  You MUST have at least one pf_user setting.

  Example:

  pf_user bob secret 1.2.3.4

- pf_header

  ...

  Example:

  pf_header

- pf_isinterval

  ...

  Example:

  pf_isinterval

- pf_domain

  ...

  Example:

  pf_domain

- pf_header

  ...

  If the pf_header setting cannot be found in the message to be sent out, POPFetch looks for the 'X-Rcpt-To: ' header, followed by the 'To: ' header, then if even that header cannot be found it will send the message to the default address, if that does not exist then the message will be appended to the duds file, duds.dat

  pf_header is not case sensitive

  handles multiple addresses, over multiple lines in header lines

  Example:

pf_header

- pf_max_duds

  ...

  The max_duds setting has a default of 10, and will not take settings below 1.

  Example:

  pf_max_duds

- pf_dflt_address

  ...

  If a message fails to be delivered to multiple recipients then POPFetch will only send one message to the pf_dflt_address, with a paragraph added to the top of the message specifying which recipients did not receive the message.

  Example:

  pf_dflt_address

- pf_retry_delay

  ...

  Example:

  pf_retry_delay

Example settings in dmail.conf might be:

```
pf_user fred passwd 1.2.3.4
pf_user fred2 passwd 4.3.2.1
pf_domain local.domain
pf_max_duds 15
pf_dflt_address fallback@local.domain
pf_retry_delay 60
ras_timer 5
ras_number 1,,12345678
ras_entry
ras_username our_account
ras_password account_password
ras_domain
ras_smtpip 1..2.3.4
ras_timeout 10
```

Files used by POPFetch

Note: POPFetch attempts to use the file paths as used by the rest of DMail. If it cannot then POPfetch will simply work in the directory that the executable is running from.

So most of the files below you will find in DMail's work path as set by the work_path setting in dmail.conf.

msg.dat - temporary storage of a message.

dmail.conf - where settings are stored (DMail's configuration file).

duds.dat - all messages that fail to be delivered anywhere are appended to this file.

popfetch.log - the popfetch log file, found in the DMail log_path directory.

Downloading POPFetch

Starting with the 2.8 versions of DMail, you will find popfetch included in the DMail distribution set. So look in the temporary unpack directory, dmtemp.

NB: you do not need to use the 2.8 version of DMail, you can simply copy the popfetch executable out of the unpack directory.

If you can not find it there, then please contact DMail Support, as on some platforms we do not yet build it by default (you should find it on Windows and Linux platforms).

DMail's External Authentication Protocol

As supported by NWAuth :-)

The DMail servers allow use of an External Authentication module to provide for interaction with any type of user database. The external authentication module is run as a sub process and accepts commands from the server on stdin and returns replies on stdout.

The Simple Authentication Protocol consists of the commands that are **necessary** for the dmail system to work.

The Extended Authentication Protocol includes other commands which we recommend your authentication module should implement. These allow for much easier user administration (including adding users) via utilities like our DMAdmin (windows GUI tool) or NetAuth (web based administration tool).

Our "standard" NWAuth module also has a number of other commands (mostly only operative from the command line) which are useful for things like, debugging, converting from other systems etc. So, on this page:

---

1. Definition of Simple Authentication Protocol

The application that is the external authentication module must read commands from standard in and write replies to standard out. The commands and replies defined allow the same system to be used by both DPOP and DSMTP. Note some parts of the reply are not needed for DPOP.

The only three input commands are:

1. `exit`
2. `check username password [fromIPaddress]`
3. `lookup username`

(lookup - is used when drop file path is needed but user is not connecting so the password does not need checking)

The replies are

1. `+OK`
2. `+OK  username  drop_file_path  uid  [info]`
   or
   `-ERR  reason  (less than 100 bytes)`
3. `+OK  username  drop_file_path  uid  [info]`
   or
   `-ERR  reason  (less than 100 bytes)`

NB: UID can be given as 0 on systems which don't require a uid, similarly the drop_file_path can be given as 'config'. They can not be left out.

Command Details:

exit:

The exit input tells the external authentication program to shutdown. It should respond with +OK.

check:

username: The username with optional prefix or suffix to indicate the user's domain.
This is generally in the form that the user is to be known by in the user database.
The main part of the name is taken from the username that the user used to login to DPOP or from the username of the destination address that DSMTP is looking up.
DMail adds on '@domain' as a suffix to the username if you have set the dmail.conf setting, authent_domain true.
The domain name that it uses is taken from the dpop_host setting or the first host_domain setting if the user belongs to the main domain. If the user belongs to a virtual domain then the domain is taken from the matching virtual domain line, vdomain, in the configuration file.
See notes below.

password: The password that the user used to connect to DMail (may be lowercased if lowercase_password is set in dmail.conf)

fromIPaddress: The IP address from which the user connected is optional, but is sent by DPOP.

lookup:

The lookup input is as per the check input. It is used when the drop file path is needed but the user is not connecting, so the password does not need to be checked.

Response Details:

+OK userdetails

-ERR reason

+OK: Implies command was successful. It will be returned by itself for the exit command.. If a user lookup or check has been successful it will be returned with the following information about the user attached.

username: Must match that given in lookup or check command, otherwise servers report "external authentication out of sync".

drop_file_path: Full path to the dropfile, including filename, for the user OR the word 'config'. The keyword config indicates that the servers should work out the user's drop path from the relevant settings in dmail.conf (i.e. drop_path + hash_spool or vdomain + hash_spool + drop_prefix + vdomain separator).

uid: User ID, can be a number or a name or zero, but it must not be left off. If a number, then UNIX uses that uid number when it checks the ownership of the drop file. If a name, then DPOP/DSMTP looks up that name in the UNIX password file and uses the corresponding uid. If uid is the number 0, then the drop file ownership is not checked or set.
Note: this manual used to erroneously say that the uid could be left off if no info field was returned.

info: Any additional fields that you wish to have.

These should have the syntax,

```
field1_name="value1" field2_name="value2" ...
```

See Info Fields supported by DMail

-ERR

reason

The error reply, giving a reason, which is less than 100 bytes in length.

---

2. Notes on the Definition

1. The authentication module should only ever return one line, in response to any query. The only exception to this is the search command in the extended protocol where a number of +DATA lines are returned.
2. The @domain suffix to the username is only sent by DPOP and DSMTP if the configuration file contains line: authent_domain true

3. The returned drop file path must contain the full path and filename of the drop file for that user.
4. If any directory hashing is required it must be included in the returned drop_file_path string.
5. The info field is not used by DPOP (only by DSMTP) and may be left blank if not required.
6. The uid field must always be returned.
7. If the normal drop file path specified in the configuration file is to be used then the drop file path in the response may be replaced by the single word config
8. The uid is used to set/check user ownership of drop files.
9. The uid returned may be numeric or an alphanumeric username.
10. If no uid checking/setting for drop files is to be used then the uid returned can be 0.
11. Currently the protocol does not support spaces in usernames. If you require this then please contact DMail Support.
12. The maximum length of any response is 1000 characters in total.
13. The @domain passed as part or the username can often be ignored. It is to allow for the use of virtual domains where a single DPOP server responds to connections to several IP addresses. In this case two users with the same username but who read their mail from different domains may connect to DPOP. See virtual domains for more details.

---

3. Info Fields supported by DMail

The servers of DMail support the following fields (field_name="value") in the 'info'

- Field Name: fwd
  Example: fwd="$USER,fred@domain2.com,|d:\dmail\robot.c"

  This field specifies alternate destinations for any messages accepted by DSMTP for the looked up user. This field must be returned in response to the lookup command as DSMTP is the only server to use it.

  No fwd field or an empty fwd field (fwd="") indicates that usual delivery shall occur.

  Multiple destinations can be specified using comma separation.

  Any destination specified indicates that the original destination shall NOT receive the message. Use the key destination, '$USER' as one of the destinations in the comma separated list to indicate that the original recipient should still receive the message.

- Field Name: quota
  Syntax: quota="<num>[K|k|M|m]" (default in bytes)
  Example: quota="5000k"

  This is new to version 2.7m. Returning this field in response to a lookup command tells dsmtp to use the specified individual quota for the user looked up.

  The quota will only be applied if the quota system is turned on with the setting, user_quota.

This setting overridden by any setting specified in the users _inf file.

No quota field or an empty quota field indicates that DSMTP should use the either the default quota value as set by the user_quota setting or the individual limit set in the user's _inf file.

---

4. Definition of Extended Authentication Protocol

- -DEAD [message]

  In addition to the +OK and -ERR responses the authentication module can now also return the -DEAD response with an accompanying message (version 2.5d and above).

  This response should be returned if for any reason the authentication module is temporarily unable to respond. For example if the user database to which the module connects is down.

  If DSMTP gets this response, it gives the sender a 400 level response (e.g. 450 Command RCPT User not OK. User database is down.), rather than a 500 level response to indicate that they should try sending again later. DPOP simply does not allow the user to login and passes the [message] back, so you should make the message something that you want the user to read.

- `set user [password [info]]`

  Response:

  ```
  +OK  [message]
  or
  -ERR  [message]
  ```

  or -DEAD [message]

  The message is optional, nwauth states, "user x added to database"
  Use this command to ADD or MODIFY a user.
  NB: NO warning is given if you are overwriting an existing user.
  If the password is given as '(NULL)' then the users info field information is altered only (the password is not changed).

- `mod user info`

  Response:

  ```
  +OK  [message]
  or
  -ERR  [message]
  ```

  or -DEAD [message]

  Use this to modify a user's info fields without needing a password. The message is optional, nwauth states, "user x added to database"

- `del user`

Response:

```
+OK  [message]
or
-ERR  [message]
```

or -DEAD [message]

Use this to delete a user from the database. The message is optional, nwauth states, "+OK Deleted user successfully"

- `search  string`

Response:

```
+DATA  username  info
+DATA  username  info
...
+OK Search Complete x items found
```

Notes:
string can be any piece of text found in the username or info fields (not in the password field).

The string '*' if supported indicates that ALL database entries should be listed. Other than that the wildcard character is not supported.

In version 2.0s of nwauth we have added,

```
search  string  [-from <x>] [-max <n>]
```

where -max limits the maximum number of search results to be returned to n and
-from causes results to be displayed starting at the x'th match.

---

5. Other commands that NWAuth supports

NWAuth supports all of the commands above run in interactive mode but also it supports them all run in command line mode, e.g.

```
\dmail\nwauth -lookup bob
+OK bob config 0
```

The syntax is generally,

```
nwauth -command cmd1_param1 cmd1_param2
```

Run

```
nwauth -help
```

for more information.

Extra commands at the command line (i.e. not supported in interactive mode):

nwauth -encrypt user/all (encrypts the password of a given user, or encrypts all passwords in the nwauth.txt file not already encrypted).

Extra switches at the command line:

nwauth -size x -... (sets max size of nwauth.add)

nwauth -log -... (turns on logging to nwauth.log)

nwauth -sleep x y (testing option, sleep for y seconds first and every x th response)

Notes on what NWAuth supports/does not support:

- NWAuth currently only ever responds with +OK User deleted successfully to the del command, if it cannot edit its nwauth.add file then it does not respond at all!

---

6. Examples

NWAuth is our example of authentication module code, see NWAuth Code . You will also find the version of nwauth.c in your distribution set which matches your version.

Set:

```
\dmail\nwauth
set bob password
+OK bob added to database
set fred pass fwd="$USER,bob"
+OK fred added to database
exit
```

Set: command line mode

```
\dmail\nwauth -set bob password
+OK bob added to database
```

DMail Frequently Asked Questions:

No. 1 question:
How do I set up a 'HotMail' type system?

Questions:

Answers:

1. Drop users:

   You have a few users who check their mail using a normal POP client but leave the mail on the server and want to be able to access the drop files directly, with pine for example. But DPOP converts the drop files to its own format for more efficient manipulation, so once the mail has been checked there is nothing left in the drop files and the users can't see their mail. This is easily remedied by adding a line to your dmail.conf configuration file. It should look like this:

   drop_users ralph,bill,*smith

   This would force DPOP to leave all the email messages for ralph, bill and anyone with a usercode finishing with the word smith, in drop files. Be careful not to put spaces in the list and avoid making it too general as there is a performance hit in keeping messages in drop files, that's why DPOP avoids it in the first place. This setting is only needed for users who check their mail with a POP3 connection AND leave it on the server AND want to read it with software that directly reads the drop file.

---

2. What operating systems is the DMail package available on?

   It is our intention to make it available on all common operating systems. Initially available on Linux, Solaris, HPUX and Windows NT. Please ask if you need it for another system soon.

---

3. What is the maximum number of email clients which can be handled by DPOP?

   This basically depends on the server hardware it is to run on and the type of license you buy. It is intended to be very scaleable and

to work well on large and small systems. Because of its design both large numbers of concurrent users and large numbers of email user accounts have relatively little impact on the process size and performance.

---

4. We have our own special username/password routines. Can these be used with DPOP/DSMTP?

   Yes, DSMTP and DPOP can be configured to use an external authentication process for checking username/passwords.

---

5. Is the source available so that we can tailor it to our needs?

   No, but this should not be necessary as most aspects of DSMTP DList and DPOP can be easily configured. They can also use an external password checking routine, an external routine to indicate where drop files are and how the path is hashed. DPOP can also generate statistics which can be used by an external routine for generating charging information. If there is some other aspect which you need to be able to tailor please let us know.

---

6. We would like to try DPOP but are paranoid about upsetting umpty thousand users. How can we ease into it?

   Email is a vital service so even if the current popper you are using is slow it is still a scary step to move to another one. You can't afford to upset users. So how do you ease into it. There are a number of strategies which can be helpful here.

   - If you have the luxury of a spare machine obviously installing DPOP on that first will help. It at least allows you to check out the various options you might want to use and get used to how they work. The DMSetup wizard will help you to remove it from the test machine after your testing is complete. The de install option tries to err on the conservative side. It tells you where the files are that you might want to delete. It will only remove something that is definitely part of DPOP and not any other popper.
   - If you have not got a spare machine or you have tried that and are now more comfortable but still cautious: The next easy step is to install DPOP on the main server BUT get it running on a different port. This way you can leave your original popper running. For example you might set DPOP up on port 1100 instead of 110. To do this, follow the normal installation procedure but say no to the question: "Shall I comment out current POP3 entries in inetd.conf". Then edit dmail.conf file and change pop_port line as shown below:
     pop_port 110
     pop_port 1100
     You can then get individual users to try switching to DPOP use by changing the setting in their email reading software to read on another port. This is straightforward in Pegasus mail, more difficult on some other email clients. For Eudora on Windows 95 just edit the Services file in the windows directory to change POP3 port. You can even allow someone to connect both ways although if they are

going to do this AND leave unread or undeleted mail on the server you must put a line in dmail.conf to tell DPOP to change their bin files back into a drop file at the end of each session. This should only be done if they NEED to read their mail from Unix command line or some other non DPOP connection. It will slow processing down. If Bob,Bill and Bert are Unix gurus who read their mail from the Unix command line and using a POP3 client, you might add one of the following lines to dmail.conf:
drop_users B*
drop_users Bob,Bill,Bert
Once you have run DPOP in this mode for a while you can switch back to the real POP3 port by changing the pop_port line in dmail.conf and then issuing the Tellpop reload command.

- Alternatively you can take the plunge and install DPOP directly on your main server in some off peak time. Test it with a few test accounts and if there are any problems that look difficult, revert to the previous popper. To do that all you need to do is put the lines in inetd.conf back how they were and get inet to reload. The DMSetup wizard can do this for you. If the accounts you have tested have undeleted or unread mail left on the server these must be converted back to drop files. This must be done before stopping DPOP by using either:
tellpop drop_all
to do all accounts that have used DPOP or
tellpop drop Bert
tellpop drop Bill
etc. to deal with user accounts one at a time.

---

7. Should I use username suffixes or multiple IP numbers for virtual domain support?

   Multiple IP numbers has the advantage that the users do not need to change their username setting in their email client packages. Username suffixes save you having to configure your server machine to respond to multiple ip numbers.
   The two schemes work as follows:

   If a vdomain setting line has an IP number like 1.2.3.4 in it then DPOP checks what ip number the user was connecting to and does stuff based on matching vdomain lines. If the vdomain setting line has a suffix string rather than an IP number in the same place (e.g. /xusers) then when users connect to DPOP and sends user fred/xusers DPOP picks up the /xusers and uses that to match a vdomain line. The suffix is stripped off and the prefix is added just as it would be for an ip based vdomain. From then on the two systems are the same. The other question is what do we end up with as a drop file name.

   Consider the two vdomain lines:

   - vdomain abc 1.2.3.4 xdomain.com /var/spool/mail/xdomain

- vdomain abc /xdom xdomain.com /var/spool/mail/xdomain

If a user connects to 1.2.3.4 or uses a username fred/xdom Then the Unix username used will be

- abc_fred

and the drop file used will be

- /var/spool/mail/xdomain/fred

Some mail transport systems find it easier to deliver to a drop file

- /var/spool/mail/xdomain/abc_fred

To allow for this another setting has been added

- drop_prefixes true/false

if this setting is true DPOP will use the second form for the drop file name.

---

8. Can I setup a 'HotMail' like system using DMail or DMailWeb? (Technical details on WAdduser)

   Yes, we have a Web Based Email system that offers Auto Account Creation. For general information on such systems see, Setting Up Web Based Email System with Auto Account Creation
   Our OLD way of doing this is presented below...

   Yes, using wadduser instead of NetAuth you need:

   - cwmail (web to mail interface)
   - dmail (dsmtp,dpop)
   - nwauth (external authentication module for dmail)
   - wadduser (example web cgi for adding users using nwauth)

   Note: You no longer have to use WAddUser with our new product NetAuth.

   DMAIL comes with source and binary examples of nwauth and wadduser, you should examine the source and modify wadduser.htm so that it only allows the users to automatically create their own accounts (it has extra functions which you would not want them to be able to do)

   Technical details:

   1. Fetch the source for nwauth/wadduser. This should come with dmail but if you have an earlier version you can download it from ftp: //ftp.netwinsite.com/pub/netwinsite/dmail/nwauth.zip
   2. Make any changes to the source that you want (not required)
      See How do I add extra fields to wadduser?

for some more information on this.

3. Building wadduser.cgi and nwauth (only needed on UNIX)

   Unix:
   gcc wadduser.c nwauth.c -DNOAUTHMAIN -o wadduser.cgi
   rm nwauth.o (so you can build it without NOAUTHMAIN defined)
   gcc nwauth.c -o nwauth
   Note: if you get crypt errors you may need to add, -lc -lcrypt to the end of each gcc line.

   Windows:
   Create two console (command line) projects,
   1 builds nwauth.exe from nwauth.c,
   2 builds wadduser.cgi from both wadduser.c and nwauth.c but you need to define NOAUTHMAIN as a preprocessor definition.
   NB:In both projects you will probably need to add wsock32.lib to the list of standard linked libraries.

4. Install the cgi script and the html form

   windows:
   copy wadduser.cgi \inetpub\scripts (or wherever your web server cgi directory is)
   copy wadduser.htm \inetpub\wwwroot

   Unix platforms:
   cp wadduser.cgi /home/httpd/cgi-bin (or wherever your web server cgi bin directory is)
   cp wadduser.htm /home/httpd/htdocs

5. Test the cgi, use netscape and reference your web site:
   http://your.web.server/wadduser.htm
   Fill out the form and press one of the buttons, if it fails, you will probably need to modify the 'action' in wadduser.htm

6. Tell dmail to use nwauth for user authentication, add or change in dmail.conf (/etc/dmail.conf or \winnt\system32\dmail.conf)

   authent_method external
   (unix) authent_process /usr/local/dmail/nwauth
   (NT) authent_process c: /dmail/nwauth.exe
   authent_number 1

7. Modify wadduser.htm so it only allows the actions that you want users to be able to perform, (e.g. not delete or search)

8. On UNIX you will need to set some file protections:

   touch ..../cgi-bin/adduser.log

chown nobody .../cgi-bin/adduser.log
touch /usr/local/dmail/nwauth.txt
chown nobody /usr/local/dmail/nwauth.txt

9. If you wish add a bulletin message to DPOP that welcomes all new users.
10. You can add a file, added.htm, in your cgi directory and wadduser will display the contents of the file when a user has been successfully added - underneath the 'Adding User' title.

---

9. **I want all domain1 email which does not go to a specific user to go to one designated user.**

   The setting you want is fallback_address, e.g.
   fallback_address domain1 default@domain2

   FYI . . .
   I gather that you were using forwarding rules to try to do the same thing instead of using the fallback address. I note that from the lines you had set up, you seemed to be expecting DSMTP to stop looking through the list of forward rules when it found the first match. So for example you had something like,

   forward bob@domain1 bob@domain2
   forward fred@domain1 bob@domain3
   forward *@domain1 default@domain4

   and expected DSMTP to only action the bob@domain1 line if a message came in for bob@domain1, i.e. you wanted the *@domain1 line to 'catch' any messages that did not match the first two forward rules.

   The way DSMTP has been written, all forwarding rules that are found to match for an incoming message are applied and forward rules are also applied instead of delivering the mail to the original recipient. So if a message came in for bob@domain1 given the dmail.conf lines above, bob@domain2 would receive the message AND so would default@domain4 (because both of the forward rules can be matched) BUT bob@domain1 would not receive the message.

   Whereas the fallback address setting,
   fallback_address domain1 default@domain4
   does what you want. I.e. if a message came in for bob@domain1.com and it could not be delivered, because the user database did not have an entry for bob and there wasn't a setting (forward rule, alias etc.) sending the mail to someone else, then DSMTP would deliver it to the fallback address, default@domain4, instead of bouncing the message back to the sender.

   Note: DSMTP's action of applying all forward rules is a nice feature that you will probably use for other situations.

10. **What is Relaying?**

    Sending mail to non-local users is referred to as 'relaying', as DSMTP must relay the message to the user's local SMTP server (often their ISP's SMTP server) so that it can write the message to the user's drop file (mail file on the server).

    The message may be relayed several times from server to server until it reaches the final SMTP server where the user is a local user - at least that is the theory. Because of spammers, most SMTP servers severely restrict what relaying is allowed to occur. So the message normally only gets relayed through an intermediary SMTP server if the server the email client gives the message to for sending is setup to gateway mail to another server, i.e. pass all its mail onto that server for delivery. An SMTP server set to gateway mail is often used to allow mail to be sent through fire walls.

---

11. **How do I add extra fields to wadduser?**

    To add extra fields in wadduser.htm for storing more information about the user, you will need to do the following:

    - Add the input text boxes and their appropriate variables in HTML to wadduser.htm (or the pages that you want them on)
    - Modify the source of the CGI wadduser (wadduser.c) so that it records the information given
    - Recompile wadduser.c (which requires linking to nwauth.c)
    - Replace wadduser.exe in your cgi or scripts directory with your new version

    The page that calls the wadduser CGI (wadduser.htm) has a form on it that calls the CGI as its action to perform when it is submitted, i.e when one of the buttons is pressed. E.g. action="http://server.com/scripts/wadduser.exe" calls the wadduser cgi from the scripts directory on the server.com web server. The CGI works out which of the buttons on the page was pressed and carries out the appropriate action.

    The function below web_add (from wadduser.c) is called when you click on the "add" button on the example wadduser.htm page.

    The form also has a number of variables that are passed to the CGI as part of the action of submitting the form, e.g. name, username, password. To add more fields you need to add more such input fields to the web page, in this form,
    <input type="text" name=" username" size="20">

    So to add a field to get the person's hobby, you could add to wadduser.htm
    <input type="text" name="hobby" size="20">

    Then you need to decide what you want the CGI to do with the information in the fields that you add.

    The three lines in the function below,

```
fprintf(f,"%s|",form_find("phone"));
fprintf(f,"%s|",form_find("fax"));
fprintf(f,"%s|",form_find("comments"));
```

search the form that is submitted by the wadduser.htm page for the fields, phone, fax and comments and if it finds them then it prints them into the log file, adduser.log. If it cannot find them, for example if there is no such input field on the web page (this is the case with the example wadduser.htm - there are no input boxes for phone, fax and comments) or the user has not entered anything in the box, then it will simply enter an empty string.

So to make wadduser log the person's hobby entry, you could add this line below the three above,

```
fprintf(f,"%s|",form_find("hobby"));
```

The function below ONLY writes the username, password and name entries to the nwauth.txt password file, but it writes to the log file, adduser.log, a whole bunch of input fields that don't exist. Note that nwauth only takes three fields, 'username', 'password' and 'other'. It is the 'other' field into which you can add your own fields. The function below adds the field 'name' into the 'other' field in the following format,

name="the person's full name"

The 'other' field can take as many fields as you want (until the information reaches the BFSZ definition, when you will get buffer over flows!) simply make sure that each field has the correct format and that they are separated by a space.

So to make the CGI write the hobby field onto the end of the 'other' field in nwauth.txt you should change the line in the function below from,

```
  sprintf(bf,"name=\"%s\"",name);
```

to

```
  sprintf(bf,"name=\"%s\" hobby=\"%
s\"",name,form_find(hobby));
```

This will result in nwauth.txt lines like,

bob:a234h6:name="Bob Smith" hobby="ping pong"

for the username bob, which has a password of something we cannot read as it is encrypted, and a full name of 'Bob Smith' and a hobby of 'ping pong'.

```
int web_add(void)
{
FILE *f;
char username[BFSZ],password[BFSZ],name[BFSZ];
char bf[BFSZ];
/* Check the user has filled in the required fields */
if (!check_value("Name","name","")) return 0;
if (!check_value("Username","username","")) return 0;
if (!check_value("Password","password","")) return 0;


f = fopen("adduser.log","a");
if (f==NULL) { printf("Could not write file\n"); return 0;}
fprintf(f,"%s|Add|",get_date());
fprintf(f,"%s|",mygetenv("REMOTE_ADDR"));
```

```
fprintf(f,"%s|",form_find("username"));
fprintf(f,"%s|",form_find("name"));
/* These are optional form elements to record */
fprintf(f,"%s|",form_find("phone"));
fprintf(f,"%s|",form_find("fax"));
fprintf(f,"%s|",form_find("comments"));
fprintf(f,"\n");
fclose(f);

ncpy(username,form_find("username"),BFSZ-1);
ncpy(password,form_find("password"),BFSZ-1);
ncpy(name,form_find("name"),BFSZ-1);

strlwr(username); /* Only allow lower case usernames */
do_header("Adding user");
printf("<pre>");
if (auth_exists(username)) {
  printf("Sorry, a user by that name already exists\n");
} else {
  sprintf(bf,"name=\"%s\"",name);
  auth_set(username,password,bf);
  showfile("added.htm");
}
printf("</pre>");
do_footer();
return 0;
}
```

---

12. Time Stamp and Time Zone problems (mostly on Linux platforms).

    NB: the Date field is normally added to an email by the email client. DSMTP only adds one if the email client has not put one on (e.g. if the message was created by DMail's sendmail stub).

    NB: In version 2.7l DSMTP was changed to add time stamps that are in local time on both the Date header if it adds one and on the Received lines. Before this it always stamped GMT on any Received headers that it added.

    If you are running a newer version of Linux (e.g. RedHat 5.2 etc.) then you may experience problems with the time stamp and timezone in the DMail servers. This is because of the difference in C libraries used to compile DMail. Examples of the problems are, the timezone being incorrectly specified or all time stamps being in GMT.

    To fix the timestamp problems, you need to use a version of DMail compiled with the newer libc6 libraries or have the below fix applied. There are other benefits to the new libraries, e.g. support for shadow passwords etc. and we have been building versions of DMail which use them since version 2.4j. So if you are running a platform that can support the newer libraries then we recommend that you download one, marked 'linux_libc6' from the main or beta download directory, ftp://ftp.netwinsite.com/pub/dmail

The alternative is this fix:
Create the proper link by executing this command.

ln -s /usr/share/zoneinfo /usr/lib/zoneinfo

(Sorry, I'm not sure what version of Unix this answer works on :-(

Also:
On many platforms the timezone information is incorrect so in dmail.conf you can define:
timezone xxxx
This controls the time zone string that DSMTP stamps on outgoing messages, to give it the form,
hh:mm:ss xxxx
NB: it does not alter the time printed, only the timezone string following it.

Some Examples:
timezone +1100 would give 11:30:33 +1100
timezone -0800 PST would give 11:30:33 - 0800 PST
timezone -0600 CST would give 11:30:33 - 0600 CST
timezone +0100 CET would give 11:30:33 +0100 CET
timezone +1200 would give 11:30:33 +1200

---

13. How can I transfer mail accounts (users) from my current email server?

    The best way to answer this is to give you some details on options for DMail and hopefully if you are able to tell DMail support about your current system then they can make relevant suggestions.

    It is worth noting first off that if the users are simply members of the operating system user database then you do not need to do anything with them - simply install DMail and it will find the users by default.

    DMail has two basic authentication options,

    a) use the operating system password list
    b) use an external authentication module

    There is one configuration file, dmail.conf, setting that sets this, authent_method

    For a this will either be,
    authent_method nt_user
    or
    authent_method unix_user
    depending on whether you are on a windows or Unix based platform.

    For b you set,
    authent_method external
    and
    authent_process path_to_program
    where path_to_program is the authentication program to run.

    Your options are:

    1. We provide an example authentication module, called

NWAuth, which is fully functional and is very efficient with large numbers of users.

2. You can also write your own to link to any type of user database (or modify one of ours).
3. Our example module for linking into an LDAP server, LDAPAuth.
4. Our example module for linking into DNews's users.dat file, DNAuth.
5. A customer has provided us with the source to talk to a mySQL server, which DMail support can pass on to you to use or modify.
6. There is a link on the following page to an ODBC authentication module provided by another customer, http://netwinsite.com/dmail/utils.htm

So one of the above might be an option, but it does depend on how the user's details are stored. Our NWAuth module can also be run from the command line, e.g.
set user password info="details"
so it may be possible to write a script to run that for all of the users out of your current user database or from a user list.

See the following sections in the manual for more details:
External Authentication
LDAP External Authentication
NWAuth External Authentication

14. **How can I have some users who can connect direct to DPOP but others who can only connect with DMailWeb/CWMail?**

Q:I want to have two different types of users. I want one group to have both pop and web access to their mail, and I want the other group to have web access only. How would I set this up? Would I need to run two seperate servers? I plan to authenticate using an external authentication module (talking to a MS SQL 6.5 database).

A:Yes, you can run two separate servers or you can make an external authentication module flag some users as being only allowed web access.

The trick is that DPOP only has the ip_address that the user connected from to know if the user has connected from CWMail or with another email client direct to the POP server. DPOP passes this ipaddress to the external authentication module.

So,
1. If you run two separate servers then you can use the user_ip_address setting on one of the servers to only allow connections to that server from the ip address of the cwmail machine. Each server then either needs its own authentication database or you need an external authentication routine for each server which cannot 'see' the other server's group of users in the database.

2. The nicer way is to make your user database have a flag for each user to say whether they are allowed to connect directly to the POP server or not, and then make your external authentication routine check this flag, and reject the connection if they have not connected from the appropriate IP address. The IP address that the user connects from is given in the authentication request by DPOP, e.g.
check username password ipaddress

So your authentication routine needs to check the "direct dpop connection allowed" flag and if it is false, it should check the ipaddress passed against your CWMail server(s)'s ip address and only allow the connection if it does not match. This is an example - you do not necessarily have to do it this way. The fact that the connection from IP address is passed to the external authentication module is the important point.

If I have not pointed it out before we also have the source code to another customer's SQL authentication module which I can give to you if it would help.

For more information contact
support- dmail@netwinsite.com

15. **How can I check what aliases I have set up for a user?**

Q:If I send a message to user x, how can I check what aliases are set up for that user?

A:To do this you should send a message to that username and then check the log file for lines with the word "chain" in them to see where it has been forwarded to.

You need to set,
log_chain true
in dmail.conf and then issue the command,
tellsmtp reload

You probably don't want to bother the user with a message, so you should make use of the tellsmtp command,
tellsmtp scriptfile.msc
to initiate a message to the user, but pull out before sending any data.

E.g. here is a scriptfile, bob.msc, that does this for a user bob
****
HELO domain.com
Mail From: <test@domain.com>
Rcpt To: <bob@domain.com>
QUIT
****

Once you have run the tellsmtp script (on debug log_level), then you can 'grep' or 'find' for lines with the word, 'chain' in the log file, dsmtp.log.

The following is a transcript of such an operation - looking for aliases and forward rules for the user bob.

C:\dmail>tellsmtp bob.msc
220 domain.com DSMTP ESMTP Server v2.5d
Send (HELO domain.com)
250 domain.com. Hello domain.com (161.29.99.1)
Send (Mail From: <test@domain.com>)
250 Command MAIL OK
Send (Rcpt To: <bob@domain.com>)
251 Command RCPT OK
Send (QUIT)
221 Command QUIT domain.com Service closing transmission channel to domain.com Send (QUIT)

C:\dmail\log>find "chain" dsmtp.log

---------- DSMTP.LOG
26/04 11:53:40 *** Starting rcpt chain for bob
26/04 11:53:40 *** Adding <|\dmail\drespond.exe \message.txt -subject whatever -from "root@domain.com"> to rcpt chain
26/04 11:53:41*** Adding bob to rcpt chain

Which shows that the message is delivered to the robot '\dmail\drespond.exe . . .' and to the user, 'bob'

Note: The log lines with the word 'chain' in them were only added, in version 2.5d, so if you are using a version of DSMTP older than that then you will need to grep for something like, 'process' and work a bit harder to interpret the results :-)

---

16. **I'm getting a Read Failed 109 error message, what's that?**

Q:Dpop.log is showing the error message 'Read Failed: 109', what's that?

A:The 109 error says that a "pipe has broken". The two things in dpop that use pipes are external authentication processes and dslave processes.

Most likely it is the external authentication process causing the problem, and it is probably occurring on the read that DPOP does after sending the 'exit' command to the external authentication. I.e., DPOP has told the external authentication to quit but does not get a response from it. So it checks to see if the external authentication has responded every so often (you will see the 109 error in the log every time that it does) until the timeout period is reached and DPOP gives up.

So this suggests that the external authentication routine is either not returning,
+OK\n
(+OK with a carriage return at the end) when it receives the exit command, or that it does not flush the output.

NWAuth has at times done both of these things. So you should probably upgrade NWAuth to a version from the 2.5d or higher distribution set (NWAuth 2.0b).

Note: To upgrade just nwauth you need to copy the nwauth

executable file over your old nwauth file, e.g. on NT, \dmail\nwauth.exe. You will need to stop DPOP and DSMTP first so that they stop all their NWAuth processes.

If you have your own authentication module then you should check that it does both of these things. Contact support-dmail@netwinsite.com if you have questions or a problem with this.

The other possibility for the error is that one of the dslave processes is no longer alive when DPOP thinks that it should be. If you do a tellpop status command it will show the number of slave channels that it thinks are running.

If this happens just once then it is probably not a problem, but if it continues to happen then it obviously does become a problem.

If the slave_number setting is above 0 then DPOP should always be running at least one slave process. Versions of DPOP before 2.5g had a problem with the dslave processes finding the dmail.conf configuration file, so if you cannot start a dslave process from the command line then this may be the problem. It will be evident in the log file, dslave.log (which itself may be being written to a strange directory on your machine - it is best to use a search to find it).

---

17. Can I filter messages based on the attachment name?

    There is no direct setting to filter by attachment filenames, but I believe that it can be done!.

    In the manual on our site(link below) under common optional settings you can find a setting
    msg_filter < filename>

    This points to a file which you create as just plain text and into which you can enter very basic filtering rules.

    But let's say we wanted to filter emails with the attachment filename of 'happy99.exe'

    We could have

    msg_filter f:\dmail\filter.txt

    and in filter.txt

    reject body begin 0666 happy99.exe
    reject body Content-disposition: attachment; filename= "happy99.exe"

    These two rules should pick up the required messages. The first reject rule is for uuencoded attachments and the second rule is for the more common MIME encoded messages.

    The rejection rules are done on simple string searches, so we suggest that you send a test message with an attachment to yourself and open up the drop file in a text editor. From this you can identify for yourself this text within the body of such messages. You will then be able to refine your rules to catch the

type of attachments your users get.

You will no doubt find the command,
tellsmtp filters
useful as it lists all filters found,and their number which corresponds with the rule number given in the line logged when a filter is matched by an incoming message.

NB: you cannot use wildcard characters in body filter rules!!!

reject body *.vbs
will not work, you should have,
reject body .vbs
in order to be a little less general we suggest,
reject body .vbs"

You can use wildcards in header processing filters - DSMTP uses a different sort of processing for them, because they are shorter and therefore do not need to be processed so efficiently.

There is another problem to the suggestion above. Sometimes an email client might split the,
Content-disposition:...
line on to two lines. In which case the suggested filter will not pick it up.

The suggested filter above is still worth adding, but we are working on a MIME parser which extracts all the MIME details so that attachment filtering and other filtering will become much easier.

Please contact DMail Support for an update on when that will become available.

---

18. Tell me about the SMTP protocol?

    The SMTP protocol is the way that an email client talks to an SMTP server in order to send a message. Note: Often it is two SMTP servers talking to each other (relaying), rather than an email client and a server.

    A typical SMTP transaction looks like (this is NOT an RFC example),

    client: (opens TCPIP connection to port 25)
    server: 220 tosh.com DSMTP ESMTP Server v2.5f
    client: EHLO tosh.com
    server: 250-tosh.com. Hello tosh.com (161.29.2.46) < cr>
    250-ETRN<cr>
    250-DSN<cr>
    250 HELP
    client: MAIL FROM:<bob@tosh.com>
    server: 250 Command MAIL OK
    client: RCPT TO:<tam@tosh.com>
    server: 250 Command RCPT User found OK
    client: DATA
    server: 354 Command DATA Start mail input; end with < CRLF>.<CRLF>
    client: From: bob@tosh.com

client: To: tam@tosh.com
client: Subject: hello
client:
client: this is the message body, line 1
client: line 2
client: .
server: 250 Command DATA Processed mail data Ok
client: quit
(server drops TCPIP connection)

Notes:

- The client sends EHLO rather than HELO if it is capable of Extended SMTP (ESMTP) Protocol
- The server advertises all of its ESMTP capabilities if the client opened with EHLO
- In the DATA stage the client sends all of the message headers and then a blank line and then the message body. It sends a dot on a line by itself to indicate that it has finished.
- If the ESMTP client wants to send a message body line with just one dot on it then it should 'dot stuff' and send two dots and the DMail servers know how to handle this.
- If the client wants to be notified of the message delivery (not reading confirmation which is handled by the receiving email client) then it can specify a DSN. E.g. MAIL FROM:<bob@domain> NOTIFY=FAILURE

  Where FAILURE could be, NEVER, FAILURE, SUCCESS and/or DELAY. See Bounces and DSNs and also RFC1891

To send an email message without a client (and to enable you to try out SMTP protocol) you can create script files (filename.msc) for DSMTP and run them with tellsmtp.

Note: For the definite word on SMTP please search for the SMTP RFC on the internet (RFC821).

---

19. How do I add Multiple IP numbers on a single machine?

    Windows NT: (workstation 4)

    You need to edit the properties of your TCPIP Protocol to add the new ip address to your network card (NIC).

    Go to the Network settings section of the Control Panel, select the Protocol Tab, and then select TCP/IP Protocol and click the Properties button.

    You will be presented with the Microsoft TCP/IP Properties dialog window. On the IP Address tab, click on the Advanced button.

    Select the network card (NIC) to which you wish to add the ip address. Then click on the Add button and enter the new IP address and the netmask for your network (if you don't know your netmask copy the one for the other ip address - a reasonable guess is 255.255.255.0).

Unix based platforms:

It is fairly easy to add multiple IP numbers for a single machine, up to 255 per interface is fairly straightforward. 1024 is usually possible with minor patches. The exact method varies from one form of Unix to another see http://www.nethelp.no/net/vif/readme.html for more information.

As an example on Linux you would do the following:

```
su - root
ifconfig eth0:2 999.59.4.31 up
```

to add a second ip number 999.59.4.31. The number :2 can be anything between :1 and :255

---

20. Can I specify a RANGE of IP addresses?

    For most settings in dmail.conf that take an ip address, you can specify a comma separated list of entries (no spaces after the commas as a general rule) and you can also specify a range or wildcard.

    We DO NOT gaurantee that you can use all of them for every setting, but we do try to code with this flexibility. So if you are wondering if a setting will take a range for example then try it out, don't just expect it to work :-)

    NB: If a setting is a 'restrictive setting' then to get through the restriction a value must get through all the restrictions in the comma separated list.

    Here are some examples:

    NB: Some of the examples in this FAQ were incorrect. Fixed 23 May 2000.

    NOTES:
    '!' indicates NOT
    '*' is a wildcard (generally for use at the start or end of a string, but with ipaddresses can be useful in the middle)
    '?' is a single character/digit wildcard
    'x-y' is a range from x to y (including x and y)

    NB: you can use, '!*?' OR a range, you **can not** use both, so this is not allowed,
    user_ip_address *,!1.1.1.0-255 (**bad**)

    The examples use the setting user_ip_address which restricts what ip addresses can connect to dpop.

    1. user_ip_address *,!161.29.5.24
    allows all ip addresses to connect, except 161.29.5.24

    2.
    user_ip_address *,161.29.3-5.24
    allows the following ip addresses to connect,
    161.29.3.24
    161.29.4.24
    161.29.5.24

    3.

user_ip_address *,!161.29.5.*
allows all ip addresses to connect, except,
161.29.5.0
...
161.29.5.255

4.
user_ip_address 161.29.3-5.0-255
allows the following ip addresses to connect,
161.29.3.0-255
161.29.4.0-255
161.29.5.0-255

5.
user_ip_address *,!161.29.*.24
allows all ip addresses to connect, except,
161.29.0.24
161.29.1.24
161.29.2.24
...
161.29.255.24

6.
user_ip_address *,!161.29.20?.24
allows all ip addresses to connect, except,
161.29.200.24
161.29.201.24
161.29.202.24
...
161.29.209.24

Note with this last example, if an ip address was, 161.29.009.24 then it would be allowed to connect.

---

21. I want to UPGRADE, ... ?

    An upgrade is in general a quick and simple procedure. The same utility that you used to install DMail, dmsetup, has an upgrade option that does it all for you.

    Note: we are always very careful when making changes to our programs that we do not 'break' them for existing setups. Having said that it is an easy thing to do so upgrading is not something we recommend doing whenever you feel like it - "don't fix what isn't broken" if you like. You should take particular care when upgrading from a version that is much older than the current beta version (e.g. 6-12 months).

    Things to consider when upgrading the DMail server (or a part of it):

    1. See the updates page,
       http://www.netwinsite.com/dmail/updates.htm
       to see which version you wish to upgrade to. If you are not sure then contact dmail support to confirm the version you should upgrade to. This applies particularly to versions out of the beta directory of the FTP site,
       ftp://ftp.netwinsite.com/pub/dmail/beta

Note: you can if you wish only upgrade one of the servers or utilities from the dmail distribution set - if you are after a particular feature in a recent beta release then this is often a good option.

2. Download the distribution set from our ftp site, ftp://ftp.netwinsite.com/pub/dmail

   If you are ftping from a command line then login as the user 'anonymous' and provide your email address as a password, then cd to pub/dmail.
3. Save a copy of your configuration file, dmail.conf (typically \winnt\system32\dmail.conf or /etc/dmail.conf)
4. You may want to revert back to your current version, so just in case you should try to save a copy of each of the executables that your system uses. If you have your last distribtion set then that should be enough. If not then you should save each of the server directories, e.g. \dmail (typically contains dpop, dsmtp), \dmail\dwatch, \dmail\dlist.

   DMSetup will not touch any of your critical data.

   For Your Information ...
   The critical data for your email server is almost all in the mail drop file and bin file directories, (defaults are, \dmail\in and /var/mail). The upgrade will not touch these directories, but of course if you wish to back them up then that is never a bad idea.

   The other critical information to think about is:
   a) mailing list information (lists.dat and users.dat for each list) - stored in the dlist directory which should be fairly small to back up.
   b) If you run external authentication then your user data base may be in a directory which dmsetup works in. NWAuth stores the user database in the dmail directory in nwauth.txt and on newer versions in nwauth.add as well.
5. Shutdown the DMAdmin windows GUI tool if you have it open (dmsetup can't upgrade dmadmin.exe if it is running).
6. Unpack the distribution set and run the utility dmsetup.
7. DMSetup should detect that you already have DMail installed and offer the upgrade option (2). DMSetup will stop each of the servers and then copy the new versions of the executables over the old ones. It will also upgrade your manual pages, *.htm in the dmail directory. Once it has finished upgrading it will ask you if you want it to start the servers again.
8. You should now check that the new version is working. You should at least,
   a) send a message through the system and,
   b) if you use dlist, post a message to a mailing list.

If you suspect that something has not upgraded, then you should attempt to manually stop that server or program and then run dmsetup again.

If you have problems then please do contact dmail support .

---

22. I want to MOVE DMail, ... ?

    Moving DMail to another machine is a fairly easy procedure. Here is a suggested method to help you remember the most common things. Each setup will be different so think if there are any other things that you need to copy over for your setup.

    Note on License Keys:

    > Your DMail license key was created for your old machine's specific machine name, e.g. server1.your_domain.com (UNIXish machines) or SERVER1 (Windows machines).
    >
    > If the new machine has the same name as your old one then simply load your key into the new machine with the tellpop command,
    > tellpop key xxxx-xxxx-xxxx-xxxx-xxxx
    > at the point below where you have started DPOP.
    >
    > If the new machine has a different name, then you need to email our Sales department, sales@netwinsite.com for a replacement key. You need to tell them the name of your new machine. They should email you your new key within 48 hours (usually only 24 hours).
    >
    > If you don't yet have your new key, do not worry, when when you start dsmtp it will create itself a temporary trial period key. So it should start and work straight away for you.

    Suggest Method for Moving DMail ...

    1. install the same version of dmail on the new machine but don't start the server when the installation utility asks you if you want the servers started
    2. copy across to the new machine your dmail.conf file typically /etc/dmail.conf or \winnt\system32\dmail.conf
    3. Copy over any other files included into dmail.conf or referenced in it, e.g. alias files.
    4. Edit your host_domain settings in dmail.conf (and your dpop_host setting) so that your new machine name is included **at the end** of the list of host_domains (also known as synomyms)
    5. now if it won't impact on your old server, start the new server up and try sending a few test messages through it

    Once you are ready to switch completely to the new machine ...

    6. Stop all servers on both machines
    7. Copy over the mail drop files, e.g. /var/spool/mail or \dmail\in

       NB: if your bin_files and _inf files are in other locations don't forget to copy those as well.
    8. Copy over the work_path directory, e.g.

/usr/local/dmail/work or \dmail\work

9. Check dmail.conf on the new machine to see that all directory paths exist and that you have copied over any necessary things
10. Start up the new server and monitor it for the next few hours.

If you have problems then please do contact dmail support .

---

23. **I want to park mail for a domain (but mail is rejected as no relaying)**

    The setting that you need is,
    relay_to etrn_domain
    so that DSMTP will always accept mail destined for the domain etrn_domain.

    Then dsmtp will accept the mail and park it when it cannot connect to the server.

    It will try to send it every 2 hours and bounce it after max_retrytime hours (default is 2 days).

    When the connecting email server sends the ETRN command dsmtp will try to send all mail addressed to that domain in its queue.

    The other setting that you can use to bypass the DNS record if you have problems is,
    gateway etrn_domain ipaddress
    so that dsmtp uses the ipaddress given rather than doing a dns lookup on etrn_domain.

    In versions 2.8e and above, we added a new setting to DSMTP for that can also help with this. It is suspend_domain, e.g.,
    suspend_domain fred.com
    This setting stops DSMTP from processing any queue files destined for this domain, unless specifically requested by an ETRN commmand. So it is a good setting to use if someone will not be collecting their mail for a period of time longer than max_retrytime. NB: it can also be a bit dangerous to use for that same reason.

    In 2.8e we also added the setting, etrn_relay which allows all servers in a server farm or load sharing arrangement to receive an ETRN command sent to just one server.

24. **Can I run DSMTP (and DPOP) on another port?**

    Yes, the setting that you want is,
    smtp_port 1025
    then restart dsmtp (with DMAdmin or on UNIX platforms with,
    tellsmtp shutdown
    /usr/local/dmail/dm_start.sh
)

    Similarly for dpop,
    pop_port 1110
    (/usr/local/dmail/dpop_start.sh to start dpop on UNIX).

NB if you are using dmadmin then you will have to select a new host to monitor with the following syntax as the ip address,
127.0.0.1:1025:1110:
so that it looks for the servers on the correct ports.
(you may need to set the password for this to work, with,
tellpop pass xxxx
,where xxxx is the password)

25. **Can I delete queue files from the queue?**

    Yes, you can delete or move them with the result that that message is not delivered, however there is a big BUT...

    Currently if you move queue files out of the work directory (work_path) you cannot easily put them back in. You can copy a queue file back into the work_path directory and dsmtp will pick up on it the next time it reaches that queue file number. But dsmtp may have created another queue file of that same number, so if you overwrite it then that message will be lost.

    Also note that some queue files will be in use by dsmtp and so locked. The tellsmtp status command gives you information on what queue files are in use.

    More information: See the section on Queue Files in the Disk Use and Files section.

---

26. **What things can I do to secure my mail system against hackers?**

    Here is a list of things that we can think of. If anyone has suggestions or gets hit by a hacker please let us know so that we can add to this list.

    - In general use ssh when sending root password across internet
    - Use fake_vrfy, so that dsmtp responds falsely to checks on usernames on your system
    - Use smtp_welcome (version 2.8a and above only) to hide what SMTP server you are using, and what version it is.
    - Set manger_ip_address to limit manager commands to coming from as small a number of ip addresses as possible
    - Use the tellpop password command to set your manager password to something secure
    - Use shadow password files, which dmail supports when authent_method is set to unix_user (linux users use libc6 download).
    - Check what UID your 'robots' run as, see Robots running as root - **Security Note**
    - If a hacker is trying to guess passwords you will see a lot of the following messages in dpop.log on info log_level,
      Info: Rejected bob, authent said bob password wrong or not a valid user
      So you can search for the keyword, 'Rejected' in dpop.log

27. **Does CWmail and DMail server support multi-threading?**

    Yes and No. I will explain.

    First DMail:

    DMail is made up of an SMTP and a POP server, DSMTP and DPOP. Both of these servers are mostly just a single process and thread, so they would only run on one processor at one time.

    They have been written to be extremely efficient, and we believe that these servers are more efficient because of their single process architecture.

    However there are two 'bottle necks' for single process mail servers. To overcome these both servers can spawn subprocesses. Both DSMTP and DPOP spawn subprocesses for doing the user authentication, and DPOP also spawns a subprocess to 'burst' drop files, if a user's drop file is bigger than a certain size.

    So these subprocesses can be run on different processors to the main server processes.

    So Yes, DMail can take some advantage from a multiprocessor system, but it is not written as a threaded process.

    NB: it is worth noting that the biggest 'bottle neck' for an email server is the disk access times. Hence we recommend spending more money on fast disks rather than a multiprocessor environment.

    RE: CWMail

    CWMail is a CGI, as such CWMail runs as a single process spawned by the web server on practically every click on the web pages that it displays. So it depends on your choice of web server as to how worthwhile it is to run on a multiprocessor environment, but in general because each instance of the CGI running is a separate process in the OS environment, there should be no problem.

28. **Is there a limit to the length of a username?**

    Yes, there is. DPOP limits you to 78 characters in the username (this includes the domain name if you have set authent_domain true). So if your domain name was 10 cahracters in length, then you are limited to usernames of maximum length, of 78-1-l0 = 67 characters for local usernames.

    DSMTP does allow longer usernames because it needs to be able to relay on messages to people with longer usernames.

    NB: if you are using external authentication then the response that the module returns is not allowed to be longer than 1kbytes in total. So you will have to limit your length of username to something sensible, so that there is room to return long fwd="" fields for mail redirection.

    So if you impose your own limit of say 40 characters, you should not have any problems.

29. **Running DMail on your ISP's Server**

    We are often asked if it is possible to run DMail on an ISP's server.

    Basically the answer for DMail is no. The DMail server needs to be run with root privilege and in most cases a box can only run one Mail server.

    You can run DMail on your ISP's machines, if they are not already running a mail server on that box, or they provide you with a box at their site, for which you have root access.

    It may be an option for you to run a 'downstream' server on a local box of yours and have your ISP relay mail for your domain to you. DMail can send the ESMTP ETRN command to collect mail for such a domain.

    You may also be able to get your ISP to forward all your mail to just one POP mail account. Then the use of DMail's POPFetch is an option.

    Separate to the question of DMail is whether you can use one of our Web Based email CGIs such as CWMail on your ISP's 'virtual web server'. Please see the following FAQ for information on this,
    http://netwinsite.com/dmailweb/faqs.htm#Q18.

30. **Robots running as root - Security Note**

    Q:> We have customers who would like to forward e-mail into external programs.
    > However, we have had to disallow this because we noted
    > that Dmail was running these external programs as root.
    > How can we tell dmail not to run external programs as a priveledged user
    > and will this break auto-responders and mailing lists?

    A:If DSMTP can work out a user's uid (e.g. from the /etc/passwd file or from the authentication module response) then it will run the 'robot' as that user's uid.

    In the case of the question I think that our NWAuth authentication module is being used. It responds with lines like,
        +OK username config 0
    where the 0 on the end is the user's id. It returns 0, i.e. root, for ALL users.

    Also, up until version 2.8l if DSMTP could not work out a user's uid then it would run the robot as the same user as itself - i.e. root!

    This means that it is **important to restrict use of robots**, e.g. NetAuth only allows users to set the text of the autoresponder robot.

    On Windows machines it is not as common to allow access to users to create robots, but if it is allowed then the same issues need to be considered.

    Here are some options ...

1. modify your authentication module to return a user id, e.g. that of the 'mail' user.

2. We are adding setting,
robot_defaultuser <userid> <password - NT only>
which defaults to root if not defined.
If set then dsmtp overrides anything returned by the authent module so that all robots are run as the specified uid. If set to -1 then no robots are run. This should be available in 2.8l to be built 8 Jun 2000. It will apply to UNIX based **and** Windows platforms.

The DMSetup utility will add it by default on fresh installation in 2.8l onwards and prompt users to add it on upgrade.

You should specify a user with this setting that does not have any more privilege than it needs.

On UNIX platforms DMSetup will default this setting to the 'mail' uid, and you will probably want to create a special robot user with far less privilege. On Windows platforms DMSetup will set the setting to 'ROBOT_USR robot_usr' by default (i.e. username and password the same) and the sysadmin will need to create this account - probably in the Guest group.

3. Currently we have the domain_chroot setting, e.g.,
domain_chroot domainone.com /usr/local/robots
which makes all robots on the specified domain run with a root directory of, /usr/local/robots. I don't think that the robot can access outside of that with root access, but there may be clever trickery that hackers know.

4. you control what programs the users run via a web gui. E.g. drespond is an example of this. NetAuth controls who can run drespond and what options it is given.

RE: mailing lists and autoresponder

Mailing lists are not affected as DList handles these and is a separate process.

The Drespond robot is affected, but with all of the options above there is no reason why they cannot keep working. You may simply have to make copies of the executable in the domain_chroot directory etc.

31. Can I use DMail for a Remote or Dial Up Mail Server?

    Yes, DSMTP can be a remote or dial up mail server.

    Options:

    - DSMTP sending ETRN command to upstream Mail server (may be using RAS dialup):

      Setting the ras_timer makes DSMTP send the command, ETRN domainx.com, to the upstream server at the specified interval. DSMTP will send ETRN commands for all of your 'local' domains (as set by your host_domain or vdomain settings).

      The upstream server will then send all mail for those

domains as soon as it can. Since your server is online it should be able to send the mail through to your local DMail server.

This is probaly the option to choose if you are retrieving mail for an entire domain or a number of domains.

See the links in the ETRN section for more information .

- Running POPFetch alongside local dsmtp for retrieving mail:

  POPFetch runs on the local mail server machine. It will periodically dial up your upstream server and collect all mail waiting in specified POP accounts. It will then process those messages and separate them out for individual users on your domain. It will feed the messages to the local DSMTP server so that it can deliver them locally.

  Often you can get whoever is running your upstream server to collate all mail for you into one POP mailbox for POPFetch to retrieve, e.g. in DSMTP this is easily done with the dmail.conf setting,
  forward *@yourdomain bob@domainx.com

  Follow this link for more information on POPFetch.

Note on Dynamic IP addresses:

If the machine where you want to run the Mail server does not have a Static IP address then you are probably limited to using POPFetch.

Some ISPs can support receiving an ETRN command for your domain when you are on a Dynamic IP address. It is not typical that they can as it requires specific dynamic DNS support,so you cannot infere that they are a sub-standard ISP for not offering it:-)

Note on bounces:

Using ETRN is a better option than popfetch if it is important that people sending mail to your local accounts receive 'bounce messages'. Most mail servers will try to deliver mail every few hours for a specified period if they cannot reach the final destination (your server) on the first go. At the end of that period, typically 1-2 days, they will 'bounce' the message back to the sender. With POPFetch (and some ETRN setups) the upstream mail server will consider the mail delivered once it recieves it (because it wrote the mail to a POP account). So if your server does not collect the mail for a long time (and nobody notices) then the sender would not be notified. ETRN can suffer from the same problem - so you should check with the upstream provider if it is a worry to you.

32. Can I use DMail from behind a firewall or proxy server?

    In most circumstances yes, but there are some circumstances where you may need to rely on an 'outside world' SMTP server.

    NB: we are using the term 'firewall' loosely. We will mostly talk as if you are running a Proxy Server on your firewall box, rather

than a router.

There are two main things that you need to provide,

1. DSMTP needs some way to connect to a DNS server to resolve domain names to IP addresses.

2. DSMTP needs some way to connect directly to the outside world SMTP servers for non-local mail delivery.

Here are some options, (Option 4 will soon be our recommended solution)

1. **Run DMail on the firewall box itself (so not really behind the proxy at all)**

   For some firewalls you won't be compromising security greatly to run the proxy server on the firewall box so that mail bypasses the proxy. In most cases if doing this you would store all mail on the firewall box until it was collected by the local email clients. You could store the mail on a network drive if you had a file server for example, but in most cases you would probably not do this because setting up the network drive connection would lessen the security of the firewall box.

2. **Relay via a DSMTP Server on your firewall box (bypass the proxy server)**

   The idea here is that the two DSMTP servers, one on the firewall box lets call it A, and one behind the firewall box (B), can pass on to each other the messages that each can not deal with. In this way the DSMTP server on the firewall allows mail to bypass the proxy server but no mail is stored on the firewall box.

   Outgoing mail will be 'gatewayed' from B to the firewall DSMTP server A which has access to the non-local SMTP servers and the DNS server(s) for non-local mail delivery. So A 'relays' mail for B.

   Incoming mail will arrive at DSMTP server A which will 'gateway' all local mail to DSMTP server B.

   To do this you need to,

   1. Tell server B to gateway ALL outgoing mail to server A
   2. Tell the firewall server A to accept outgoing mail for 'relay' from server B
   3. Tell the firewall server A to accept incoming mail addressed to local domains on B
   4. Tell the firewall server A to gateway incoming mail addressed to 'local domains' on to B

   So if a.a.a.a is the ip address of server A and b.b.b.b is the ip address of server B...

   On server B add to dmail.conf,
   gateway * a.a.a.a

On server A add to dmail.conf,
forward_from_ip b.b.b.b
relay_to domain1.com
relay_to domain2.com
gateway domain1.com b.b.b.b
gateway domain2.com b.b.b.b
(keep adding relay_to and gateway settings for all local domains)

See also, Routing.

3. **Gateway all outgoing mail to an Outside world SMTP server (via the proxy server)**

   You can avoid most problems by **'gateweying'** all outgoing mail to an SMTP server in the outside world, that provides you with 'relay' access.

   This is similar to the option above in that outgoing mail is relayed via an SMTP server with 'outside world access', but with this option, mail goes through the proxy server, and incoming mail comes **direct** to your proxy server.

   To do this you add a setting to dmail.conf like,
   gateway * x.x.x.x
   where x.x.x.x is the ip address of your firewall server.

   The possible problem with this is that you need to set up the proxy so that,

   A. anything connecting to port 25 from the DMail server address is mapped to port 25 at your ISP's SMTP server IP address.

   B. anything connecting to port 25 from other addresses (e.g. outside world ones) is mapped to port 25 on your DMail server's IP address.

   Some proxy servers are not capable of this type of setup on the single port (25), and some will do it 'automatically' with a 'SMTP proxy' feature. If you are using a router then it will probably have no problems with this.

   If your proxy cannot do that sort of setup, then note that in version 2.8n we have altered the gateway setting so that you can specify the port on the proxy,
   gateway * x.x.x.x:1025
   This allows you set up up two port mappings on the proxy,
   1025 -> ISP_IP_Address:25 (for outgoing mail)
   25 -> DMail_IP_Address:25 (for incoming mail)

   You also **must** get whoever is running the outside world server to accept mail from your server for relaying. ISPs by default will stop you from relaying through their box unless you have their permission (it is to stop them being abused by spammers). They will probably do this based on the ip address of your proxy server - as that is the address that mail from your DSMTP server will appear to them to have originated from. If they are running DSMTP then they would add the forward_from_ip setting for your ip address.

4. **Proxy DNS Access AND use telnet proxy to reach non-local SMTP servers**

   Sometimes people have their own DNS server behind or on the firewall, but for most people they don't so you have to,

   Set up a proxy server to relay all DNS lookups:

   Doing this varies between proxy servers. It is important to note that DNS lookups can be done on a TCPIP port and/or a UDP port. So you need to set up your proxy server to at least relay TCPIP connections on port 53 to port 53 on the DNS server. On most proxy servers you can setup a TCPIP 'port mapping' or 'link' to do this.

   You also need to tell DSMTP which DNS server to use by adding the dmail.conf setting,
   dns_host y.y.y.y
   where y.y.y.y is the ip address of the DNS server to use. You **must** restart DSMTP after changing or adding this setting.

   Using telnet proxy to reach non-local SMTP server:

   You cannot simply add a 'port mapping' for port 25 on most proxy servers and expect them to 'proxy' all incoming and outgoing connections on port 25 to/from the DSMTP server.

   When the DSMTP server tries to reach a non-local server it is trying to connect to that server directly on port 25. Even if we added a setting to DSMTP to make it connect to your proxy server, there is no way for the proxy server to map an incoming connection on port 25 to the required server which could be anywhere in the world!

   So we have recently added a new setting to DSMTP (in version 2.8n) which makes it open all non-local connections via your proxy server's telnet port.

   Because there is no fixed syntax for proxy telnet ports the new setting allows you to specify the connection string to be given to the telnet server, e.g.
   destination_ip:25

   The setting is,

   proxy_domain <wildcard_domain_name> ip[:<port>] <proxy_request_string [optional macro $IP]>

   where $IP is the resolved IP address of the destination domain, E.g.,
   proxy_domain * 1.2.3.4:23 $IP:25
   where 1.2.3.4 is the ip address of your proxy server. This example results in all outgoing mail being sent to the telnet proxy at 1.2.3.4, where the proxy server takes a request string of, x.x.x.x:25. DSMTP will replace x.x.x.x with the DNS resolved IP Address of the the destination domain.

5. ---

Does DMail support CDONTS?

No, but there is now an option in DMail to deliver mesages written to file.

I am afraid that CDONTS were created too much as part of the web server/email server combination and do not use the standard SMTP protocol that they 'should' for sending mail. So as far as I know there is no way for CDONTS emailing calls to get the mail message to the SMTP server.

However it would seem that it is an option (possibly the default) for CDONTS calls to write email messages to a given directory.

We have recently added a feature whereby DSMTP will 'pick up' messages written to file in a directory and deliver them to the destination address specified in the message headers in the file.

So given that you can somehow make your system create such files on the server's local drives, then dsmtp can deliver them.

For information on the setting needed and the message file format see the DSMTP Settings List,
spool_dir.

NB: you need a 2.8 version of DSMTP so I suggest that you download the latest 2.8 build (probably 2.8v) from the directory,
ftp://ftp.netwinsite.com/pub/dmail

NB: This new feature has not been thoroughly tested yet and we can not be sure that it will handle the file format created by CDONTS. So contact DMail Support if you strike any problems or need us to make changes to the system.

6. ---

My Users are not appearing in the nwauth database file?

Often people are mistaken about the way that nwauth stores usernames and other data, so here is an explanation.

When you add a user to nwauth, e.g. by running it at the command line

nwauth
set bob secret
quit

then nwauth will write the username and the details to the file,
nwauth.add
in this format,
username:password:blah
where 'blah' is any other information you store for the user.

When you modify a user's details, nwauth simply adds

another line for the same user to nwauth.add with the new password or other details.

When you delete a user, nwauth adds a line like,
username:(DELETE):(DELETE)
to the nwauth.add file.

When the nwauth.add file reaches a certain size nwauth will delete that file and update the main database file, nwauth.txt. When it updates nwauth.txt it processes it in order, so in general it uses the last entry for a user found in the nwauth.add file and deletes the user if it finds a line for a user with the '(DELETE)' password. It does this so that all of its operations are instantaneous no matter what size the user database is.

Often you will only have an nwauth.add file, and the nwauth.txt file will not appear for several days.

If usernames are not being added to the file here are some helpful hints:

- Look in the nwauth.add file not the nwauth.txt file
- Try nwauth from the command line. See EAP definition for details of the commands.

  If it works from the command line, then you probably have the incorrect setting in dmail.conf or netauth.ini. This is now, authent_process for both dmail.conf and netauth.ini. (On NT use a drive letter or UNC name when specifying the process, e.g. c:\dmail or \\machineA\cdrive\dmail rather than just, \dmail which is ambiguous).

  If it still fails then see the next suggestion below.
- **Is nwauth modifying the nwauth file in the directory you think?**
  This might be the problem if you are running nwauth across a network or on an NFS drive.

  If you are suspicious of this then search your machine for any copies of nwauth.txt or nwauth.add.

  NWAuth decides where to find/create the nwauth.add and nwauth.txt files in one of two ways.

  1. It looks at the **local** dmail.conf file and uses the value of the dsmtp_path setting, typcially c:\dmail\

  2. You run it with the command line argument, -path, to specify the path to use, e.g. at the command prompt,
  c:\dmail\nwauth -path c:\dmail
  or in dmail.conf or netauth.ini,
  authent_process c:\dmail\nwauth -path c:\dmail

  NB: you should not need to set the path unless you are running in a server cluster. We don't recommend that you use the -path option unless you need to - i.e.

be careful of using it as a quick fix without understanding why it is not working without it. Talk to DMail Support if you want help working out why it is not working.

- **There could be a file permission problem:**

  (See also, Authentication for DMail and NetAuth on Clustered machines and Network Drives)

  On NT:
  nwauth is spawned by dsmtp and dpop, which are spawned by dwatch service which is typically running as the 'System Account', so check that the directory that nwauth is running in and the nwauth files give full access to that user.

  If using NetAuth, note that it is generally being run as a specific user by the web server. You to work out what the user is (typically IUSER_XXXX, where xxxx is you machine name). Then ensure that that user is created on the box and has the permissions needed to run nwauth and create/access the nwauth.add and nwauth.txt files in the dsmtp_path directory.

  On UNIX:
  nwauth is spawned by dsmtp and dpop, which may be spawned by the dwatch process. All of these will be running as root, so in general you should not get a problem. If you are running nwauth on an NFS then you will probably need to set root access on the file share so that these programs can access it.

  During installation the NetAuth binary should have had its s bit (sticky bit) set. It's ownership should also have been set to the root user. This is so that the web server will always run it as root.

  Unless the permissions are set as such then NetAuth will not be able to function properly.

  So,
  ls -l netauth.cgi
  should show something like this,
  -rwsrwsr-x root:root netauth.cgi

  If not then set these permissions with the commands...
  chown root:root netauth.cgi
  chmod 6775 netauth.cgi

  NB: with file permission problems, it is often a good idea to give all access to the user to get it working and then work backwards restricting the access to the level you are happy with.

7. **Authentication for DMail and NetAuth on Clustered machines and Network Drives**

(AKA: Running NWAuth on a shared network drive)

Most of the following is for the authentication module NWAuth, but much of it applies when using any authentication module.

When you have a cluster of DMail servers or a DMail server and NetAuth running on a web server you need to allow them to all access the same user database.

For authent modules like, MySQLAuth this is not a problem because the database is accessible via TCPIP from any machine on the network.

For nwauth and some other modules which use local database files this is a problem.

Here are 3 solutions for nwauth:

1. make all of the servers run the same copy of nwauth on a shared network drive.

2. run a separate nwauth on each server, and set the -path option so that they all work on the same nwauth.add and nwauth.txt files on a shared network drive.

3. run a TCPIP daemon that spawns nwauth on one machine and then run a 'client' for that daemon on each of the servers.

Option 3 has some good benefits, so we are creating a new module called, TCPAuth (with TCPAuth_client) to do that. Contact DMail Support for information.

Option 1 is the current option being used by customers so is known to work on UNIX and NT. Setup for option 1 is described below.

Option 2 is pretty similar to option 1, so if you want to do that read the suggestions below and you will probably be able to work out what to do.

So to recap, the information below is how to,

Run nwauth on a shared network drive.

- **For those on UNIX and using NFS drives:**

  nwauth is spawned by dsmtp and dpop, which may be spawned by the dwatch process. All of these will be running as root, so in general you should not get a problem.

  During installation the NetAuth binary should have had its s bit (sticky bit) set. It's ownership should also have been set to the root user. This is so that the web server will always run it as root.

  Unless the permissions are set as such then NetAuth will not be able to function properly.

  So,

ls -l netauth.cgi
should show something like this,
-rwsrwsr-x root:root netauth.cgi

If not then set these permissions with the commands...
chown root:root netauth.cgi
chmod 6775 netauth.cgi

You will probably need to set root access on the file share so that these programs can access it.

In both dmail.conf and netauth.ini use the authent_process setting to specify the full path to the nwauth process and pass it the command line argument, -path, e.g.
authent_process /shared/dmail/nwauth -path /shared/dmail/
(in dmail.conf the authent_method setting should also be set to, 'authent_method external')

Remember to restart both DSMTP and DPOP after changing the authent_process setting,

tellpop shutdown
tellsmtp shutdown
/usr/local/dmail/dm_start.sh
/usr/local/dmail/dpop_start.sh

If authentication fails, then look in the dpop.log file to see why. You will see at the start of the dpop.log file after restarting dpop if it has had difficulty spawning the authentication process.

- **For those on NT and using network drives:**
  1. Run the dwatch service as a specific user, e.g. IUSER_DMAIL, which you must create on ALL boxes, i.e. the mail server box, the web server box and the box that holds the network drive (it will depend on your setup how many boxes this is, it may be just 2 boxes or many more).

     Set this in Control Panel, Services. Select 'dwatch monitor for dmail servers' and click on Startup and then change the check the 'Log on as this account:' button and enter the account (IUSER_DMAIL) to be used and any details.

     You will have to stop and restart the dwatch service in the Services dialog to make this change take effect.
  2. Similary you have to ensure that the Web Server spawns NetAuth as the same user, IUSER_DMAIL, so that it can access nwauth on the network drive.

     Most web servers allow you to set the username used for spawning CGIs (that is

what NetAuth is). Often they are spawned as the anonymous user login account, IUSER_XXXX where XXXX is your machine name - look in your NT system user database for such a user.

You won't know what the password for that user is, so you won't be able to add that user to the other boxes in your cluster. This is why we suggest creating the new user, IUSER_DMAIL, on all of the boxes.

If you have the IIS server see the specific note below.

3. Use UNC names for the paths rather than mapped network drives, e.g., authent_process \\machineA\Cdrive\dmail\nwauth.exe

   UNC names allow the dwatch service which will start automatically after a reboot to reach nwauth on the other box even if no one is logged in yet. Whereas mapped drives are only accessible once someone has logged in to the box, so won't be accessible to dwatch (and hence dsmtp and dpop) after a reboot until someone logs in to the mail server box.

4. In both dmail.conf (c:\winnt\system32\dmail.conf) and netauth.ini (c:\inetpub\scripts\netauth.ini) use the authent_process setting to specify the full path to the nwauth.exe file and pass it the command line argument, -path, e.g.
   authent_process \\machineA\Cdrive\dmail\nwauth.exe -path \\machineA\Cdrive\dmail\
   (in dmail.conf the authent_method setting should also be set to, 'authent_method external')

   Remember to restart both DSMTP and DPOP after changing the authent_process setting. The best way to do this is either with DMAdmin or using the Control Panel Services dialog.

   If authentication fails, then look in the dpop.log file to see why. You will see at the start of the dpop.log file after restarting dpop if it has had difficulty spawning the authentication process.

- **Special note on the IIS web server:**

  Follow all the suggestions above. If they do not work check the following magic setting as this sysadmin did:

I just tried changing the settings in IIS.
Under Web Site properties->Directory Security->Anonymous Access...->Allow Anonymous Access[edit]

I have "IUSER_DMAIL" as the username and have set up all permissions for that user on both mail server boxes. I had ticked, 'Enable Automatic Password Synchronization'. I unticked this, and NOW IT WORKS!

Contents:

Search: Enter Word(s) to find

Submit

Free Trial Updates

License Prices

Products Downloads

Support Company

Contents:

Search: Enter Word(s) to find

Submit

Free Trial Updates

License Prices

Products Downloads

Support Company

Contents:

Search: Enter Word(s) to find

Submit

Contents:

Contents:

Search: Enter Word(s) to find

Submit

Free Trial Updates

License Prices

Products Downloads

Contents:

Contents:

Search: Enter Word(s) to find

Submit

Free Trial Updates

License Prices

Products Downloads

Support Company

Contents: - Domains

Search: Enter Word(s) to find

Submit

Contents: - Domains

Support Company

Contents:

Contents:

Contents:

Search: Enter Word(s) to find

Submit

Free Trial Updates

License Prices

Products Downloads

Support Company

Contents:

Contents:

Search: Enter Word(s) to find

Submit

FAQ - Anti-Spam

Also see the manual section on Spam.

1. What Anti-Spam measures does DMAIL have?
2. How do I stop people sending mail to non-local users?
3. How do I allow the trusted client/domain with the non-static IP address to send out mail?

Other FAQs that might relate:

1. Can I delete queue (spam) files from the queue?

---

1. **What Anti-Spam measures does DMAIL have?**

   For general information on Anti-Spam features in dmail see, Spam Rules

   The following settings all relate to Anti-Spam features:
   forward_from_ip
   relay_to
   forward_user
   fromip_max
   fromip_nolimit
   max_rcpts
   msg_filter
   ban_ip
   Note this is a non-exhaustive list - we are coming up with new anti-spam features all the time. Our support staff can help you choose the best settings for your server.

---

2. **How do I stop people sending mail to non-local users?**

   Sending mail to non-local users, relaying, is allowed by defaut by DSMTP. To restrict such relaying you should add various Relaying Restrictions in dmail.conf. See the, Relaying Restrictions

   section in the main manual.

   Note: In 2.8 versions and above, the DMSetup utility will add,
   forward_from_ip 127.0.0.1,x.x.x.*
   where x.x.x is the first 3 parts of your machine's IP address. This limits relaying to users sending from IP addresses 127.0.0.1 or that start with x.x.x .

---

3. **How do I allow the trusted client/domain with the non-static IP address to send out mail?**

   To allow this customer to relay out to the world, there are these options... (often you will run a mixture of them)

   1. Add,

forward_from_ip 1.2.3.*
where 1.2. is the first part of the ip addresses that customer connects from.

Side-effect: This creates a relaying hole for any spammer also connecting from that range of ip addresses - probably not very likely.

2. Add
forward_from their.domain.com

Side-effect: creates a relaying hole for spammers pretending to be from their.domain.com which is easy to do.

NB: if you have this setting for your main domain then Open Relay Databases like ORBS will almost certainly add you to their 'bad servers' list!

3. Add
forward_user true
which turns on the 'recentpop' or 'POP before SMTP' system. This allows users to relay mail for the default period of 2 minutes after checking for mail. You can increase this period with the setting, forward_window.

NB: this handling of this system was improved greatly in version 2.8m. It worked fine for medium sized servers but became inefficient on large servers where the window was set large, e.g. 1-2 days.

4. Add,
auth_allow relay

(version 2.8n and above)
DSMTP supports the SMTP AUTH command when this setting is added. This allows the user to turn SMTP AUTH on in their email client. SMTP AUTH means that the email client will provide the username and password (same as on POP server) to authenticate on your SMTP server when connecting to send out mail. Once authenticated the 'trusted' user is then allowed to relay.

NB: adding this setting will mean that some email clients like Netscape Mail force the users to turn on SMTP AUTH. Generally this is not a problem as Netscape Mail instructs them on how to do it, but it may be confusing to some users.

If using the forward_user system as well then you should probably set the setting,
hide_auth recentpop

We also have a new proxy widget called SmtpAuth (currently only in windows beta form) which takes a username and password to authenticate to an SMTP server with.

So users with an email client that does not support the SMTP AUTH command can run this on their machine and point their client at it instead of directly at your smtp server. It then authetnicates to your server before sending on any mail feed to it.

If it is a whole domain coming through another trusted server then they could use the SmtpAuth proxy and feed all their outgoing mail through it. As we only have SmtpAuth on

NT their server would have to be running on NT. If their server is DSMTP then we are adding a setting so that DSMTP auths all connections to a given ip address. So they could run with that setting.

See also the Relaying Restrictions section in the main manual.

DMail also allows restrictions to be placed on the volume of messages coming from a particular IP number going through DSMTP per hour.
fromip_max: Restricts the number of messages per hour that DSMTP will accept from an IP number.
fromip_nolimit: Permits exceptions to from_ip_max for certain IP numbers. This applies to the IP number of the sender.

Message filtering is also available, though it should be used with care. DMail doesn't do logic checks of them so it may be possible to accidently reject everything (!).
msg_filter: Gives a filename containing message filtering rules. An explantion of those rules is at the other end of the link

You can also straight out ban anyone from a particular IP address from connecting.
ban_ip: Specifies an IP address that DSMTP may not talk to.

Frequently Asked Questions:
- Setting up a Web Based Email with Auto Account Creation system

No. 1 question:
How do I set up a 'HotMail' type system?

Questions:

Relevant questions in the Miscellaneous FAQ (may have different titles):

Answers:

How do I setup WAdduser? (Technical details on WAdduser)

For general information on 'HotMail' type systems see, Setting Up a Web Based Email System
technical details for WAddUser are below.

Yes, using wadduser instead of NetAuth you need:

- cwmail (web to mail interface)
- dmail (dsmtp,dpop)
- nwauth (external authentication module for dmail)
- wadduser (example web cgi for adding users using nwauth)

Note: You no longer have to use NWAuth with our new product NetAuth (but it is still a good option).

DMAIL comes with source and binary examples of nwauth and wadduser, you should examine the source and modify wadduser.htm so that it only allows the users to automatically create their own accounts (it has extra functions which you would not want them to be able to do)

Technical details:

1. Fetch the source for nwauth/wadduser. This should come with dmail but if you have an earlier version you can download it from ftp://ftp.netwinsite.com/pub/netwinsite/dmail/nwauth.zip
2. Make any changes to the source that you want (not required)
   See How do I add extra fields to wadduser? for some more information on this.
3. Building wadduser.cgi and nwauth (only needed on UNIX)

   Unix:
   gcc wadduser.c nwauth.c -DNOAUTHMAIN -o wadduser.cgi
   rm nwauth.o (so you can build it without NOAUTHMAIN defined)
   gcc nwauth.c -o nwauth
   Note: if you get crypt errors you may need to add, -lc -lcrypt to the end of each gcc line.

   Windows:
   Create two console (command line) projects,
   1 builds nwauth.exe from nwauth.c,
   2 builds wadduser.cgi from both wadduser.c and nwauth.c but you need to define NOAUTHMAIN as a preprocessor definition.
   NB:In both projects you will probably need to add wsock32.lib to the list of standard linked libraries.
4. Install the cgi script and the html form

   windows:
   copy wadduser.cgi \inetpub\scripts (or wherever your web server cgi directory is)
   copy wadduser.htm \inetpub\wwwroot

   Unix platforms:
   cp wadduser.cgi /home/httpd/cgi-bin (or wherever your web server cgi bin directory is)
   cp wadduser.htm /home/httpd/htdocs
5. Test the cgi, use netscape and reference your web site:
   http://your.web.server/wadduser.htm
   Fill out the form and press one of the buttons, if it fails, you will probably need to modify the 'action' in wadduser.htm
6. Tell dmail to use nwauth for user authentication, add or change in dmail.conf (/etc/dmail.conf or \winnt\system32\dmail.conf)

   authent_method external
   (unix) authent_process /usr/local/dmail/nwauth

(NT) authent_process c:/dmail/nwauth.exe
authent_number 1

7. Modify wadduser.htm so it only allows the actions that you want users to be able to perform, (e.g. not delete or search)
8. On UNIX you will need to set some file protections:

   touch ..../cgi-bin/adduser.log
   chown nobody .../cgi-bin/adduser.log
   touch /usr/local/dmail/nwauth.txt
   chown nobody /usr/local/dmail/nwauth.txt
9. If you wish add a bulletin message to DPOP that welcomes all new users.
10. You can add a file, added.htm, in your cgi directory and wadduser will display the contents of the file when a user has been successfully added - underneath the 'Adding User' title.

---

1. What is the Maximum number of users the system can handle?

   This depends on your system hardware and the setup of our products.

   Each of the products involved in our solution for a Web Based Email system can be registered on an Unlimited User License. Therefore our software need not limit the number of users your system can have.

   As for performance, we currently have customers operating successfully with tens of thousands of users (POP mail accounts), and the bigger ones are fast approaching 100,000 users. We know that our products perform well with these sort of numbers.

   As far as larger systems go, we can only say that we are commited to producing efficient software and as part of this we are eager to work with any customer to help them run our products on larger systems.

   It must be said that our products have been designed to be scaleable and we would anticipate that on systems over 50,000 users you would seriously be looking at running our products on multiple servers.

   We are endeavouring to do more testing on larger test setups, and we will add any results to those already on our dmail performance page,
   http://www.netwinsite.com/dmail/perform.htm

---

2. **How do I add extra fields to wadduser?**

   To add extra fields in wadduser.htm for storing more information about the user, you will need to do the following:

   - Add the input text boxes and their appropriate variables in HTML to wadduser.htm (or the pages that you want them on)
   - Modify the source of the CGI wadduser (wadduser.c) so that it records the information given

- Recompile wadduser.c (which requires linking to nwauth.c)
- Replace wadduser.exe in your cgi or scripts directory with your new version

The page that calls the wadduser CGI (wadduser.htm) has a form on it that calls the CGI as its action to perform when it is submitted, i.e when one of the buttons is pressed. E.g. action="http://server.com/scripts/wadduser.exe" calls the wadduser cgi from the scripts directory on the server.com web server. The CGI works out which of the buttons on the page was pressed and carries out the appropriate action.

The function below web_add (from wadduser.c) is called when you click on the "add" button on the example wadduser.htm page.

The form also has a number of variables that are passed to the CGI as part of the action of submitting the form, e.g. name, username, password. To add more fields you need to add more such input fields to the web page, in this form,
<input type="text" name="username" size="20">

So to add a field to get the person's hobby, you could add to wadduser.htm
<input type="text" name="hobby" size="20">

Then you need to decide what you want the CGI to do with the information in the fields that you add.

The three lines in the function below,

fprintf(f,"%s|",form_find("phone"));
fprintf(f,"%s|",form_find("fax"));
fprintf(f,"%s|",form_find("comments"));

search the form that is submitted by the wadduser.htm page for the fields, phone, fax and comments and if it finds them then it prints them into the log file, adduser.log. If it cannot find them, for example if there is no such input field on the web page (this is the case with the example wadduser.htm - there are no input boxes for phone, fax and comments) or the user has not entered anything in the box, then it will simply enter an empty string.

So to make wadduser log the person's hobby entry, you could add this line below the three above,
fprintf(f,"%s|",form_find("hobby"));

The function below ONLY writes the username, password and name entries to the nwauth.txt password file, but it writes to the log file, adduser.log, a whole bunch of input fields that don't exist. Note that nwauth only takes three fields, 'username', 'password' and 'other'. It is the 'other' field into which you can add your own fields. The function below adds the field 'name' into the 'other' field in the following format,
name="the person's full name"
The 'other' field can take as many fields as you want (until the information reaches the BFSZ definition, when you will get buffer over flows!) simply make sure that each field has the correct format and that they are separated by a space.

So to make the CGI write the hobby field onto the end of the 'other' field in nwauth.txt you should change the line in the function below from,

```
  sprintf(bf,"name=\"%s\"",name);
```

to

```
  sprintf(bf,"name=\"%s\" hobby=\"%s\"",name,form_find(hobby));
```

This will result in nwauth.txt lines like,
bob:a234h6:name="Bob Smith" hobby="ping pong"
for the username bob, which has a password of something we cannot read as it is encrypted, and a full name of 'Bob Smith' and a hobby of 'ping pong'.

```
int web_add(void)
{
FILE *f;
char username[BFSZ],password[BFSZ],name[BFSZ];
char bf[BFSZ];
/* Check the user has filled in the required fields */
if (!check_value("Name","name","")) return 0;
if (!check_value("Username","username","")) return 0;
if (!check_value("Password","password","")) return 0;

f = fopen("adduser.log","a");
if (f==NULL) { printf("Could not write file\n"); return 0;}
fprintf(f,"%s|Add|",get_date());
fprintf(f,"%s|",mygetenv("REMOTE_ADDR"));
fprintf(f,"%s|",form_find("username"));
fprintf(f,"%s|",form_find("name"));
/* These are optional form elements to record */
fprintf(f,"%s|",form_find("phone"));
fprintf(f,"%s|",form_find("fax"));
fprintf(f,"%s|",form_find("comments"));
fprintf(f,"\n");
fclose(f);

ncpy(username,form_find("username"),BFSZ-1);
ncpy(password,form_find("password"),BFSZ-1);
ncpy(name,form_find("name"),BFSZ-1);

strlwr(username); /* Only allow lower case usernames */
do_header("Adding user");
printf("<pre>");
if (auth_exists(username)) {
  printf("Sorry, a user by that name already exists\n");
} else {
  sprintf(bf,"name=\"%s\"",name);
  auth_set(username,password,bf);
```

```
  showfile("added.htm");
}
printf("</pre>");
do_footer();
return 0;
}
```

---

3. **How can I have some users who can connect direct to DPOP but others who can only connect with DMailWeb/CWMail?**

Q:I want to have two different types of users. I want one group to have both pop and web access to their mail, and I want the other group to have web access only. How would I set this up? Would I need to run two seperate servers? I plan to authenticate using an external authentication module (talking to a MS SQL 6.5 database).

A:Yes, you can run two separate servers or you can make an external authentication module flag some users as being only allowed web access.

The trick is that DPOP only has the ip_address that the user connected from to know if the user has connected from CWMail or with another email client direct to the POP server. DPOP passes this ipaddress to the external authentication module.

So,
1. If you run two separate servers then you can use the user_ip_address setting on one of the servers to only allow connections to that server from the ip address of the cwmail machine. Each server then either needs its own authentication database or you need an external authentication routine for each server which cannot 'see' the other server's group of users in the database.

2. The nicer way is to make your user database have a flag for each user to say whether they are allowed to connect directly to the POP server or not, and then make your external authentication routine check this flag, and reject the connection if they have not connected from the appropriate IP address. The IP address that the user connects from is given in the authentication request by DPOP, e.g.
check username password ipaddress

So your authentication routine needs to check the "direct dpop connection allowed" flag and if it is false, it should check the ipaddress passed against your CWMail server(s)'s ip address and only allow the connection if it does not match. This is an example - you do not necessarily have to do it this way. The fact that the connection from IP address is passed to the external authentication module is the important point.

If I have not pointed it out before we also have the source code to another customer's SQL authentication module which I can give to you if it would help.

For more information contact
support-dmail@netwinsite.com

DMail Frequently Asked Questions: Converting To DMail (from other email systems)

Questions:

Transferring Users:

Transferring mail data - drop files

General

Answers: Transferring Users

1. How can I transfer mail accounts (users) from my current email server?

   The best way to answer this is to give you some details on options for DMail and hopefully if you are able to tell DMail support about your current system then they can make relevant suggestions.

   It is worth noting first off that if the users are simply members of the operating system user

database then you do not need to do anything with them - simply install DMail and it will find the users by default.

DMail has two basic authentication options,

a) use the operating system password list
b) use an external authentication module

There is one configuration file, dmail.conf, setting that sets this,
authent_method

For a this will either be,
authent_method nt_user
or
authent_method unix_user
depending on whether you are on a windows or Unix based platform.

For b you set,
authent_method external
and
authent_process path_to_program
where path_to_program is the authentication program to run.

Your options are:

1. We provide an example authentication module, called NWAuth, which is fully functional and is very efficient with large numbers of users.
2. You can also write your own to link to any type of user database (or modify one of ours).
3. Our example module for linking into an LDAP server, LDAPAuth.
4. Our example module for linking into DNews's users.dat file, DNAuth.
5. A customer has provided us with the source to talk to a mySQL server, which DMail support can pass on to you to use or modify.
6. There is a link on the following page to an ODBC authentication module provided by another customer,
   http://netwinsite.com/dmail/utils.htm

So one of the above might be an option, but it does depend on how the user's details are stored. Our NWAuth module can also be run from the command line, e.g.
set user password info="details"
so it may be possible to write a script to run that for all of the users out of your current user database or from a user list.

See the following sections in the manual for more details:
External Authentication Modules List
External Authentication
LDAP External Authentication

NWAuth External Authentication

2. **How can I convert from Unix_user (/etc/passwd, shadow passwords etc) to NWAuth.**

   NWAuth uses the system crypt function on UNIX based platforms so you can probably simply copy the first two columns from your /etc/passwd file into nwauth.txt. NWAuth should get password comparisons correct. You might want to use the utility cols to pull out the first two columns in /etc/passwd.

   If you are using shadow passwords or yellow pages etc (note that you should be using libc6 versions of DMail) it may not be so simple.

   Please let us know if you find anything out in this area so that we can add to this FAQ.

3. **We have our own special username/password routines. Can these be used with DPOP/DSMTP?**

   Yes, DSMTP and DPOP can be configured to use an external authentication process for checking username/passwords.

4. **Columns, A utility developed for extracting data from a file to command lines in a batch file for adding users to NWauth**

   You can download Columns from the DMail Utilities Download page.

   This utility scans a file for data and writes it out to file in a specified format. An example use is to create a batch file to add a list of users to nwauth, E.g. turn this 2 line file,

   bob;pass1;blah blah
   james:password;blah blah

   into,

   nwauth -set bob pass1
   nwauth -set james password

   Where, the 'set' command line option for external authentications adds the specified user with the given password. So running the last file as a batch file would add the 2 users to the nwauth user database.

Answers: Transferring Data - Drop Files

5. **What mail box (drop file) format does DMail use?**

   DMail uses the Sendmail 'standard' format for user drop files (mail boxes). This is where all messages for each user are appended onto the end of one file. The messages are separated by a blank line and a line starting with 'From ...' (the syntax of the From line can need to be specific).

   There is an added complication that when a user connects to DPOP to collect their mail, DPOP removes the mail from that file and converts it to its own bin file format. See various

sections in the Disk Administration section for more details.

NB: if your users run some sort of command line program that directly accesses the drop_file then you need to use, the tellpop drop command to make DPOP convert its bin files back to a drop file for that user. You can set this to be done automatically on pop logout - see drop_users.

6. **Where drop_files (mail boxes) get put (hashing) and what they are called.**

   Drop files are created in the directory specified by the drop_path setting in the dmail.conf config file for the main domain or any aliases of it (as set with the host_domain settings). If the user is on a virtual domain the vdomain setting's last parameter sets the drop file path for that domain.

   In addition to this the setting hash_spool sets whether you want any directory hashing done (extra levels of directories added in so that there are not too many files in any one directory). A hash_spool setting of 0 means that no hashing should occur. If you have hash_spool set to something else then you need to be aware of this when you copy drop files across from other systems.

   NB: we have a utility FixHash which allows you to move drop files between different hashing methods.

   The DMail servers generally name drop files with the same name as the username part of the email address. E.g. if bob has the email address,
   bob@netwinsite.com
   then his drop file name would be simply,
   bob

   If bob was on a virtual domain then by default he would have a prefix added to his drop file name (taken from the first parameter of the vdomain setting),e.g.
   domainx_bob
   This is a safety precaution to ensure that two users with the same username on different domains never get their mail merged together!

   NB: FixHash can add prefixes onto the start of drop files for you if you wish, or you can set, drop_prefix false in order to turn drop file prefixes off.

   For more details on all of this see the Disk Administration section of this manual as well as other FAQs on this page.

7. **A list of what we know about converting from other email servers.**

   Below are details provided by customers who have converted from other systems to DMail. Please let us know how you get on if your current system is not on this list, or you can add to any of the entries. NB: Don't be surprised if your current system is not on this list, it is a very new list. We have had customers change from most common email servers to DMail, so you are probably not the first :-)

   NB: One 'gotcha' is differences in drop file hashing - see Where drop_files (mail boxes) get

put. on this page.

- Sendmail/CuciPOP:
  The drop file format is the same so simply copying over drop files works. You can use the /etc/passwd file (including shadow password files) as is, by setting authent_method to Unix_user, or you can use our Columns utility to move your user database to any of the external authentication modules, e.g. nwauth, LDAP, SQL etc.
- Elm, Pine clients:
  Uses sendmail drop file format so should be no problem. Need to convert DPOP's bin files back to drop file format before using these, see drop_users.

8. **DPOP Compatibility Settings**

   See,
   msg_separator
   drop_old
   drop_kill

Answers: General

9. **Change Over Time Suggestions**

   Here are some suggestions for handling the point at which you change over to using DMail ...

   - **use the bind_in setting**

     You can make the DSMTP and DPOP servers use the bind_in setting to bind to one specific ip address. This allows you to run them alongside your existing mail server (given you bind that server to another specific ip address).

     This can be a good way to do final testing of your DMail setup, but do be aware of things like the drop_file directories (set with the drop_path setting) conflicting with the old server.

10. **I run an automated mailer, e.g. CGI or command line mailing...**

    If your current email system has provision for sending emails generated by robots or CGIs or some sort of command line mailer, then DMail will in most cases be able to work with or replace that part of your system.

    For UNIX users, when you install dmail it replaces Sendmail with what we term the 'Sendmail Stub' which replaces the command line emailing part of Sendmail, so that you can still email from the command line, e.g.
    mail bob@domainx.com

    Many people run CGIs or a command line mailer which sends emails automatically via sendmail's command line mail command - this sendmail stub allows these to keep working

when you change to DMail. (A common problem is that the replacement of sendmail does not happen on install. Our sendmail stub is only about 40k in size and dmsetup tries to put it in /usr/sbin. So check the size of that file and that any links for 'mail' or whatever point to that stub. Our stub creates a file, dsmtp_path/sendmail.log which logs if it has been called and with what arguments - so check that file to see if your CGI or mailer is actually running the stub).

On Windows platforms, most mailers of CGIs if well written will talk to the SMTP part of the email system on port 25, using the SMTP protocol. Given you have a mailer such as this there should not be any problem in continuing to use it with DMail.

Some email systems (in particular on Windows platforms) have built in support for emailing from web pages or talking to CGIs and command line mailers. This in our opinion is 'bad form' and getting away from such proprietry behaviour is an advantage of moving to DMail :-) To help in moving to DMail for such systems we are creating a CGIMail CGI to run on your web server which when fed a web page will take fields off a form on that page and create an email out of them and feed that over TCPIP to any SMTP server.

In addition to the above we are adding the ability to put a message file in a directory, which DSMTP then picks up and sends as a way of interfacing a mailer with DMail.it, using information in the file, writing CGIMail to email form information from a web page.

Contact DMail Support for more information on any of the above.

11. We would like to try DPOP but are paranoid about upsetting umpty thousand users. How can we ease into it?

    Email is a vital service so even if the current popper you are using is slow it is still a scary step to move to another one. You can't afford to upset users. So how do you ease into it. There are a number of strategies which can be helpful here.

    - If you have the luxury of a spare machine obviously installing DPOP on that first will help. It at least allows you to check out the various options you might want to use and get used to how they work. The DMSetup wizard will help you to remove it from the test machine after your testing is complete. The de install option tries to err on the conservative side. It tells you where the files are you might want to delete. It will only remove something that is definitely part of DPOP and not any other popper.
    - If you have not got a spare machine or you have tried that and are now more comfortable but still cautious: The next easy step is to install DPOP on the main server BUT get it running on a different port. This way you can leave your original popper running. For example you might set DPOP up on port 1100 instead of 110. To do this follow the normal installation procedure but say no to the question: "Shall I comment out current POP3 entries in inetd.conf". Then edit dmail.conf file and change pop_port line as shown below:
      pop_port 110
      pop_port 1100
      You can then get individual users to try switching to DPOP use by changing the setting in their email reading software to read on another port. This is straightforward in Pegasus mail, more difficult on some other email clients. For Eudora on Windows

95 just edit the Services file in the windows directory to change POP3 port. You can even allow someone to connect both ways although if they are going to do this AND leave unread or undeleted mail on the server you must put a line in dmail.conf to tell DPOP to change there bin files back into a drop file at the end of each session. This should only be done if they NEED to read there mail from Unix command line or some other non DPOP connection. It will slow processing down. If Bob,Bill and Bert are Unix gurus who read there mail from the Unix command line and using a POP3 client you might add one of the following lines to dmail.conf:
drop_users B*
drop_users Bob,Bill,Bert
Once you have run DPOP in this mode for a while you can switch back to the real POP3 port by changing the pop_port line in dmail.conf and then issuing the Tellpop reload command.

- Alternatively you can take the plunge and install DPOP directly on your main server in some off peak time. Test it with a few test accounts and if there are any problems that look difficult revert to the previous popper. To do that all you need to do is put the lines in inetd.conf back how they were and get inet to reload. The DMSetup wizard can do this for you. If the accounts you have tested have undeleted or unread mail left on the server these must be converted back to drop files. This must be done before stopping DPOP by using either:
tellpop drop_all
to do all accounts that have used DPOP or
tellpop drop Bert
tellpop drop Bill
etc. to deal with user accounts one at a time.

12. Drop users:

    You have a few users who check their mail using a normal POP client but leave the mail on the server and want to be able to access the drop files directly, with pine for example. But DPOP converts the drop files to its own format for more efficient manipulation, so once the mail has been checked there is nothing left in the drop files, so the users cant see their mail. This is easily remedied by adding a line to your dmail.conf configuration file. It should look like this:

    drop_users ralph,bill,*smith

    This would force DPOP to leave all the email messages for ralph, bill and anyone with a usercode finishing with the word smith in drop files. Be careful not to put spaces in the list and avoid making it too general as there is a performance hit in keeping messages in drop files, that's why DPOP avoids it in the first place. This setting is only needed for users who check their mail with a POP3 connection AND leave it on the server AND want to read it with software that directly reads the drop file.

13. Is the source available so that we can tailor it to our needs?

    No, but this should not be necessary as most aspects of DSMTP DList and DPOP can be easily configured. They can also use an external password checking routine, an external

routine to indicate where drop files are and how the path is hashed. DPOP can also generate statistics which can be used by an external routine for generating charging information. If there is some other aspect which you need to be able to tailor please let us know.

14. **A typical response to someone converting from Sendmail**

>Currently we store and maintain all our users in Mysql.
> We generate the
> passwd, shadow, access, aliase and virtusertable from the database for
> sendmail. Here is the key elements:
>
> * For each user,
>
> - user name is unique.
> - password is Linux style, not Mysql style.
> - all users are maintained by existing programs. Thus the user
> maintenance feature of dmail is not needed.
> - with these setting, our users have these simple set up in their email
> programs:
>
> # outgoing email server: mail.bob.com
> # incoming email server: mail.bob.com
> # incoming email account: ...their.login.name...
> # incoming email password: ...their.login.password...
>
> * For all users,
>
> - aliases to equate different names to the same user
> - aliases to provide email forwarding. We do not use .forward files as
> there is no real home directory for the users in the mail server. -
> aliases to provide lists or multiple forwardings
>
> * For virtual domains, using virtusertable,
>
> - maps domain email addresses to unique user names.
> - define default email address or user for each domain.
>
> As an on-going business, we just cannot stop the current email
> arrangement and ask all users to change their settings. How can dMail be
> programmed to simulate the above settings?
>
> We maintain our own database and do not mind if I have to create or
> modify tables to suit the dMail structure. Please give us some
> suggestion and do not hesitate to ask if there is any query.

You have 2 options.

1. you can continue to use the same system password file,passwd, shadow, access, aliase and virtusertable created from the database. You should find that there is not too much to change in converting to DMail.

2. you can run our new SQLAuth authentication module so that the DMail servers talk directly to the MySQL database.

I suggest that you start with 1 as that will be quickest and then once you are happy with that you can look to simplify your setup and take advantage of things like our proper virtual domains (rather than the virtusertable) by moving to option 2.

So for option 1, here are the things that I can think of for you to consider, in addition to what is in the Moving to DMail FAQ (i.e. this page)...

1.RE: system password file, i.e. passwd and shadow passwords

If you install DMail on a linux box it will default to,
authent_method unix_user
in /etc/dmail.conf.

This means that DSMTP and DPOP will use the standard system user calls so you should not have any problems staying with those files as your mail user database.

(Note: Because your usernames are unique and I presume they are not the full user's email address, you do not need to use the authent_domain or preserve_domain settings, which you will see talked about throughout the manual).

2. RE: access

I don't know which file you are talking about. I presume that it is for restricting who can access the server - maybe to stop you from being an open relay and other uses.

Can you send me an example of this file and tell me more about its function.

3. RE: aliases and virtusertable

DSMTP uses the sendmail style alias files (99% of features are covered), so again you should not have any problems with these. They are specified for you main domain (as set by the first host_domain setting in dmail.conf),
alias_file <filename>
and for specific domains with,
alias_file_domain domain <filename>
where domain can be a wildcard, e.g.,
alias_file_domain *bob.com /etc/aliases

We have also recently added support for sendmail's virtusertable. Previously you had to convert all entries within that table to a mixture of our dmail.conf settings,
forward
forward_cc
fallback_address

and aliases within alias files.

With beta version 2.8d, you should be able to use the setting,
virtual_user_pre <filename>
and it should work without modification.

NB: virtual_user_pre is a new beta feature so be aware that it is possible there are problems with it. Please let me know straight away if you come across any such problems.

Version 2.8d is available from the beta directory of our site.

4. general settings:

You indicate that your machine name is,
mail.bob.com
I presume that the email domain is actually, bob.com. In which case in /etc/dmail.conf you should set,
host_domain bob.com
as your FIRST host_domain setting (which tells dsmtp that it is a local domain) and as your second host_domain setting enter,
host_domain mail.bob.com
or
host_domain *bob.com
If you have other subdomains that are synonyms of the domain bob.com

I recommend that you take advantage of the free trial period to set up a test box using your configuration. If you come across any problems please send us your dmail.conf file and either the dsmtp.log or dpop.log file showing the problem from the log_path directory. NB: it is best to send us the logs created when the logging level is set to debug. To set this edit log_level to read,
log_level debug
and then reload both dsmtp and dpop with the commands,
tellsmtp reload
tellpop reload
(as required by most dmail.conf settings).

FAQ for Large Systems - e.g. 20,000 users and above

On other FAQ pages ...

- Miscellaneous FAQ: Does CWmail and DMail server support multi-threading?
- Miscellaneous FAQ: Authentication for DMail and NetAuth on Clustered machines and Network Drives

---

1. Are there any special things that I should be setting in dmail.conf?

   In general you should not need to modify anything much in dmail.conf for a system of this size - a normal installation of DMail tries to perform to its best for all sized systems!

   The only exception to this is that you should definitely make sure that you have directory hashing turned on with the hash_spool setting. For more than 20,000 users we recommend hash_spool 2 so that there are 2 extra levels of directories.

   Here is a list of other things we recommend that you look at:

   - Logs: You should increase the size and number of your log files - aim to have at least the last 2 hours of activity avaialable in the log files. The settings to use are, max_loglen, rotated_logs and max_log_size.
   - min_space (in mbytes): You should set this setting to something like 30 mbytes, so that you are notified well in advance of disk space problems. When this minimum amount of space is reached dsmtp will alert the sys. admin. and then stop accepting connections after the

disk space gets below about 90% of that value. So if the value is larger there is a longer warning time between the alert and the offline state.

2. **List of load sharing and redundancy options**

   Here is a list of most ways in which you can share the load across multiple SMTP and/or POP servers. Some are beneficial in terms of backup servers for redundancy and some are not. See, How can I share the load across multiple servers?
   for more discussion of the options.

   1. **One main server and secondary SMTP server(s):** (recommended first approach)
      Secondary SMTP server(s) give redundancy by holding incoming/outgoing mail if main server goes off line. SMTP server(s) can share the same user database, or use mirrored user databases.
   2. **One DPOP server and multiple SMTP server(s):**
      All mail delivery shared mail collection always from just one machine. SMTP servers share drop_path and user database.
   3. **Multiple DPOP servers with separate bin files and multiple SMTP servers:**
      Users must always connect to the same DPOP server or always remove mail from POP server. Servers share drop_path and user database but not bin_path.
   4. **Multiple DPOP servers and multiple SMTP servers:**
      Full load sharing where users can access any machine. Servers share, drop_path, bin_path and user database.

   NB: with all options where the user database and/or the mail storage area is shared access to these resources obviously becomes the weak point of the system.

   We don't currently have any way to mirror server where there is no physical connection between the two servers by way of a shared network drive.

3. **How can I share the load across multiple servers?**

   The easiest way to share the load across multiple DMail servers, is to run one main server and run one or more DSMTP servers as backup servers. You add secondary MX records to your DNS server for the In this way you provide reliability without the complication of sharing the mail storage area.

   We generally suggest this as a first step, as it is simple to set up and allows you to see how just one DMail server performs. It is not until you have run one of our servers that you will get a real feel for what your load sharing requirements are. This is a sensible approach as it lets you identify where the real load bottle necks are for your system and then you can address them specifically, which may or may not include running multiple servers.

   However whether you want to do that to start with or not, read on ...

   DMail does support running multiple servers, which share both the user database and the mail storage area. In the manual we term this a 'server farm' arrangement.

   You can run multiple SMTP servers and have just one POP server or you ran run multiple of both.

You simply add one extra setting to each server's configuration file to enable the sharing of the mail storage area,
lock_id xxx
where xxx is a unique ID for each server.

You then point the setting,
authent_process path_to_authent_module
at the same authentication process on each server, and the setting,
drop_path path_to_storage_area
to a shared mail storage area.

Don't forget to RESTART both DMSTP and DPOP on each machine after doing this.

On UNIX platforms the shared storage area would normally be a shared NFS drive. Our lock_id code turns on our own file locking system which enables the file sharing on an NFS drive. You might also need a shared drive for the authentication module files, e.g. for our NWAuth module you do need to share the directory where you place the NWAuth executable.

NB: We have recently made improvements to our file locking code so you should contact DMail Support to find out which version you should run if you are setting up a 'server farm'.

On Windows platforms, you need to use UNC style names (e.g. \\serverA\shared_drive\dmail) to specify paths for the settings above between the server machines for the mail storage area. You may also need to use a UNC name for the authentication module, e.g. our NWAuth module requires this. You can map network drives instead of using UNC names but UNC names have the advantage that if the server reboots then the path specified with the UNC name will be accessible, whereas a mapped network drive is not accessible until someone logs on and the mapping is created. Whether you use UNC names or mapped network drives you MUST enable sharing on the folders that the servers are to share AND ensure that the correct permission for that shared resource is given - see the note below.

NB: the DMail servers are spawned by the DWatch service on NT, which will be running as the user 'system', so that is the user that the servers and the authentication module will also be running as. You cannot give access permission to a mapped network drive for the 'system' account. So you must change the NT DWatch service so that it runs as a specific user (in Control Panel, Services, Startup) and then give read/write permission for that user on the shared directories.

4. **I want to run multiple servers, how can I route the mail to them?**

   The answer to this question is slightly different for POP and SMTP. Often you may just want to run multiple SMTP servers and just one POP server.

   In general the easiest way to do this is by using the DNS server lookup to share out the load. You can also do it by using a router in front of your mail servers.

   Below are some suggestions based on feedback from our customers...

   (NB: We don't have much experience with routers, so if anyone is using such a setup then they might like to contribute some general information here???)

Rotating DNS MX Lookup:
For the DSMTP servers you can set up a rotating DNS MX lookup. A rotating lookup works by responding with a different answer from a list of responses on each consecutive DNS lookup. So for example,
person one looks up domainx.com and gets back mail1.domainx.com
person two looks up domainx.com and gets back mail2.domainx.com
etc.

In this way the load gets distributed

One POP server for each group of domains:
For the DPOP servers it can be easy to share load on the basis of domains. To do this you ask users from one domain or a group of domains to connect to one of the POP servers, e.g. mail1.domainx.com and another group to connect to another POP server, e.g. mail2.domainx.com and so on.

NB: if you are running web based mail then you can set CWMail or DMailWeb to connect to a specific POP server - so you can do the above without the users needing to know about the separate POP servers.

If you use this method then you can actually get away with quite separate POP servers, provided the group of users always connect to the same POP server (which is easy to ensure if users always access their mail via CWMail).

5. **Can I share the load across NetAuth and CWMail as well?**

   Yes, NetAuth just duplicate on multiple systems, needs shared access to external authentication module.

   CWMail need to share workareas and template files.

   These will be updated with proper FAQs shortly ...:-) In the mean time contact, NetAuth Support or CWMail Support for details.

6. **What is your recommendation for scaling our system?**

   We recommend that you start your mail system at Step 1 below and then you will be able to scale it up as it grows through the other steps that are applicable to your situation.

   Step 1: Start with just one box running DMail + NetAuth + CWMail + WebServer no matter what your size system.

   We also recommend that you setup 3 separate domains for each of the mail servers and for the web server, e.g.
   smtp.domain.com
   pop.domain.com
   www.domain.com
   for DSMTP, DPOP and CWMail (on the web server) respectively.

Also where you have virtual domains, get your users using the virtual domain equivalents,
smtp.vdomain.com
pop.vdomain.com
www.vdomain.com

To start with all of the above domains can resolve to the same box.

Once you have users using these domains in their email client settings and web page bookmarks then it is easy for you to redirect them to separate boxes at a later stage simply by altering your DNS records.

NB: if you are worried about mail delivery redundancy you should ask someone else (e.g. an ISP) to act as a 'secondary SMTP server' for your system. This is easily done by simply pointing your secondary DNS MX entry at their smtp server. All they have to do is ensure that mail for your domain will be accepted (by their anti relay settings) and that their server will continue trying to send it for a reasonable amount of time until your server can come back online.

Then when needed here are our recommended possible further steps...(in a rough order)

... greater than 100k users ...

- **Move the WEB server to another box (away from the POP and SMTP servers).**
- **Move the DPOP server to another box (away from the SMTP server).**
- **Add a second DSMTP server on another box. Load share with a rotating DNS MX entry.**
  You will need to add a rotating DNS MX entry that moves between your first and second DSMTP servers on alternate lookups.

  You can also use the second SMTP server just for outgoing mail or for mailing lists only.
- **Add further CWMail (web) servers. Try to move virtual domains to separate web servers.**
  Often you can easily split load across multiple CWMail servers by divinding based on domain, e.g. you may put just 2 or 3 domains on each server.
- **Add further DPOP servers. Try to move virtual domains to separate DPOP servers.**

... greater than 500k users ...

- **Add a router, so that you can add further POP, SMTP and WebServers easily**

  Once you have a router on the front of your email system you can easily load share between any of the servers by running them on separate boxes.

7. Can I share the majority of dmail.conf across multiple servers?

Yes. There should not be any problem with having,
machine a dmail.conf:

lock_id 1
#include /share/dmail.share

machine b dmail.conf:

lock_id 2
#include /share/dmail.share

where /share/dmail.share is the main common config file settings (with its own #includes) on a shared NFS drive. Provided the root of both machines can read the #included file.

The dmail.conf file is only ever read by the dmail servers so does not need locking of its own.

NB: The following settings need to be in either /etc/dmail.conf or \winnt\system32\dmail.conf. I.e. the following programs parse /etc/dmail.conf themselves for the following,

dwatch:
- manager_ip_address (if not there still accept from local)
- timezone (not really needed)
- dmserver_port (not needed)
- work_path
- log_path
- dlist_path
- dwatch_path

tellsmtp:
- work_path
- smtp_port (can be set on command line)

So you would need these set in dmail.conf on each machine.

8. Do you suggest setting use_flock to false on linux??

   No, not unless you get a specific problem. It is a new setting (version 2.8d) added for use if flock does not work (i.e. causes stop errors for dpop, rather than simply failing to get a lock) on your type of NFS.

9. Are there any problems with clients that directly access the drop file??

   (We use a commercial fileserver from Network Appliance, does the linux flock (on the client end) cause any problems?

   Theoretically yes there is a problem. A linux client which accesses the drop file directly could think that it has a flock on the drop file when it doesn't and hence a mail message could arrive into the drop file when a user was reading their mail. The consequences could be that the mail message gets lost. However the chances of this are very small, and even smaller if the number of customers using direct drop file accessing clients is small.

   The problem is that we can't think of any way to stop this - let us know if you have any ideas :-) However our general feeling is that it is not really an issue.

10. **Vitual Domains on a large system in DMail, CWMail and NetAuth overview**

    > Thank you for your prompt reply. In your response, can you also give me some
    > indication how Netwin handle multiple virtual domains? We are planning to give

> our users the choice of up to 3 domain names.

Netwin's products have good support for multiple virtual domains. If anything it can be a little confusing because of all the options available.

Firstly note that if you just want to allow the user to use all of 3 addresses, e.g.
tam@domain1.co.nz
tam@domain2.co.nz
tam@domain3.co.nz

i.e. all of those addresses are valid and are the same user, then you should add domain2.co.nz and domain3.co.nz as 'domain aliases' of domain1.co.nz rather than as full virtual domains.

E.g. ,
host_domain domain.co.nz
host_domain domain2.co.nz
host_domain domain3.co.nz
in dmail.conf and in cwmail no extra settings would be needed. You would only have to ensure that the different URLs reach the same cwmail cgi.

Given you do want separate virtual domains, i.e. where the same username is re-used on each of the vdomains, then these are also easily added.

Virtual Domains in DMail:

In dmail.conf you add your main domain with a host_domain setting,
host_domain main_domain.co.nz
(you probably have already done this)
and then add the other domains as virtual domains with vdomain settings,
vdomain a b vdomain1.co.nz c
where a,b and c are specific values for that domain. Most importantly the vdomain line sets a separate mail spool directory for each domain, so that the mail is kept completely separate.

The DMail manual has details on the vdomain setting in section 8.

Virtual Domains in CWMail and NetAuth:

In cwmail and netauth the settings you currently have in their ini files are all for your 'main domain'.

You can choose to run separate CGIs for each virtual domain or you can add 'vhost-vend' sections to their ini files for the virtual domains.

The vhost sections allow you to simplify maintenance for the vdomains. They are sections of the ini file that allow you to specify any 'over rides' of the main domain for the virtual domain.

So long as you use the vhost sections to separate out the stored mail and folders for each domain, it is easy to split to separate CGIs on separate web servers at a later date if you want to.

To keep the stored mail separate you must specify a unique workarea setting for each vhost section.

Also you should use the domain name for the pophost setting rather than an ip address. The pophost setting is used to form the user's directory name, so in general it should not be changed once you have added users (although it can be done). Using the domain name means you can change the DNS record for that domain later to point to a different box without upsetting cwmail.

Also if you use a subdomain for the pophost setting, e.g.,
    mail1.vdom1.co.nz
rather than just,
    vdom1.co.nz
, then you can easily make separate domains use totally separate pop servers at a later date as an easy way of distributing load.

E.g. here is a simplified cwmail.ini file showing a main domain and two virtual domains (all sharing the same set of templates) which will be easy to separate out at a later stage if needed:

```
#******************
pophost main_domain.co.nz
domain main_domain.co.nz
workarea /usr/local/cwmail/main_domain/
templates /usr/local/cwmail/tpl/

vhost www.vdom1.co.nz
domain vdom1.co.nz
pophost vdom1.co.nz
workarea /usr/local/cwmail/vdom1/
vhost www.vdom2.co.nz
domain vdom2.co.nz
pophost vdom2.co.nz
workarea /usr/local/cwmail/vdom2/
vend
#******************
```

NetAuth's vhost sections are just like CWMail's. Here are the links to both of the manuals that have full details, http://www.netwinsite.com/dmailweb/cwmail.htm
http://www.netwinsite.com/netauth/netauth.htm

DMail Email Servers - Performance and Requirements

Currently our largest customers are running systems of 500,000 users and are growing rapidly, e.g. some are increasing at 60,000 users/month.

DMail has been designed to be scaleable, in 'server farm' arrangements.

We are committed to making sure that it grows with our customers needs. So far it is performing exceptionally well in terms of handling load for systems of this size.

Note: This page is provided to give you some pre-purchase information. It is intended to give you a feel for system requirements.

A few notes:

- Please appreciate that many items will never be filled in and contact names may never be provided due to commercial sensitivity. Netwin is very thankful to all contributors to this page.
- Commas are used to indicate thousands on some numbers, e.g.,
  12,346
  is 12 thousand 3 hundred and 46, not a fraction over 12 :-)
- Real world systems and test systems are clearly marked as such.
- By providing this information Netwin in no way gaurantees any performance quoted nor implied.
- We are continually in the process of gathering the information for this page - if your system will be of interest to others, for being big, small, fast, unusual, etc., please let us know :-)

On this page ...

Performance

System A - real world system

General:

Running Version: 2.3j
Platform: Linux 2.0.36
Total user database size: 60-70,000 -includes forward/redirects, auto responders, POP clients and webmail clients.
No. of servers: 1 running DMAIL, another 4 for webmail serving
Authentication System: External - all users are db-based.
Virtual Domains: not available (but known to have many)
DList lists: not available (but known to have many)

Machine Details:

AMD K6-2 300, with 128M ram.

Total DMail disk useage on machine: 6.98 Gbytes

- spool files: 6.6Gbytes
- dmail (logs, index files, aliases, etc, etc): 380Mbytes

DSMTP: (averaged out to per day)

- Messages received: 7,404
- Messages delivered: 5,184
- Messages sent on: 1,604
- Total in: 7590 kbytes
- Total out: 7583 kbytes
- idle 74.5%
- drop 8.6% (writing to drop file)
- que 4.3% (processing of queued messages)
- send 0.9% (processing outgoing messages)
- lookup 7.0% (looking up usernames, i.e. checking if user local)
- robot 0.5% (waiting on autoresponders etc.)

DPOP: (averaged out to per day)

- peak sessions: 79
- Rejected connections: 2,364
- Connections processed: 40,083
- Messages served: 29,432
- Kilobytes served: 480,251

- License accounts used: 45512
- External authent channels: 5
- idle 93.2%
- pass 0.1% (password lookups)
- retr 0.8% (actioning POP RETR command - retireving messages)
- drop check 26.8% (checking drop files for messages)
- burst 12.9% (bursting drop file to bin files)

Site:

www.mailbr.com.br

Any nice comments :-)

"I really appreciate the help you guys have given, and the willingness to work together with my suggestions."

Calculated Statistics (approximate):

Average disk useage per user: 110Kbytes (6.6Gb/60,000 users)

Netwin's Summary:

We are very pleased with these statistics, particularly the idle percentages for both servers, which show plenty of room for expansion.

System B - real world system

General:

Running Version: 2.8k
Platform: Linux 2.0.36
Total user database size: > 230,000
No. of servers: 1
Authentication System: External - all users are in MySQL.
Virtual Domains:
DList lists:
Server also runs: mysqld

Machine Details:

AMD K6-2 300, with 128M ram.

loadavg: usually stays around only 1

Total DMail disk useage on machine:

- spool files:
- dmail (logs, index files, aliases, etc, etc):

DPOP:

- peak sessions: 102
- Peak external authent channels used: 8
- Peak DSlave processes used: 4

(all the following are averaged out to per day)

- Rejected connections: 12,469
- Connections processed: 11,6664
- Messages served: 42,936
- Kilobytes served: 3,094,606
- License accounts used: 235,616

Site:

Any nice comments :-)

"I want to continue to praise the efficiency of your software. One of our systems handles > 230,000 users easily without skipping a beat"

Calculated Statistics (approximate):

Netwin's Summary:

System C - DSMTP Maximum Message Throughput Benchmark

The following are some results from testing to ascertain the maximum message throughput of DSMTP. The tests were designed to test the file (disk) access and queue file handling.

Test Setup:
On machine x , 10 tellsmtps send RSET-separated messages
On machine y , dsmtp version 2.8l processes them.
Machine y, is a P300, 128Mb Ram running Red hat Linux 6,

Pertinant dmail.conf settings:
tcp_max 300 (default is 200)
max_send 50 (default is 10)
max_queue 5000 (default is 1000)
hash_qfiles true (default is false)

All messages are addressed to a dummy address which machine y has been configured to gateway back to machine x which discards them without even writing a qfile or delivering them (using the bit-bucket

feature).

Results:

1. when queue processing is suspended DSMTP will accept incoming messages at about,
80 per second

2. when it is offline (i.e. no incoming messages) DSMTP can send messages at about,
70 per second

3. During the stress-test, from the first received message to finishing all processing of the last one, the net rate is about,
50 per second.

Requirements

System Requirements are hard to give, they are very dependent on what else is running on your system and also your network bandwidth. So here are some 'rules of thumb' that we work with. NB: These are very rough. Figures are for either an NT or UNIX based box unless stated otherwise.

NB: We get a lot of customers saying,
"I need to set up a system for 1 Million users",
our response to this is that unless you currently have an active user database of 1 Million users then set up a 100,000 user system, because with our product you can add in more servers at a later date, i.e. your server can grow as your user database grows. We generally recommend starting with just one box for the email server. If redundancy is a worry then run a second box with a second SMTP server only, and if you are really worried then run two full email servers. See the Large Systems FAQ for more information.

- **Processor**

  <1,000 users - P133 or higher
  <10,000 users - try a P166 if that's what you have, else buy whatever is currently available.
  >10,000 users - higher end Pentium, e.g. PII450.
  >200,00 users - start to think about multiple servers (but wait until you get there)

  NB: Multiple processors - you may not get the mail processing speed benefit you anticipate by running multiple processors (with any mail server not just ours), because the message processing is not generally the 'bottle neck' in a mail server, it is commonly the disk access speed. However they can be made to make more of a difference if you run an external authentication module, as these (you set how many with authent_number) are spawned by both DSMTP and DPOP. DPOP also spawns, DSlave processes (slave_number) for 'bursting' large mail drop files (mailboxes).

- **Ram**

  1,000 users - 64MB RAM on NT, 32Mb on UNIX based platforms
  10,000 users - 128MB RAM
  50,000 users - 256MB RAM

>100,00 users - 512MB RAM

- **Disk**

  < 10,000 users - use whatever you have got
  >10,000 users - buy the best you can afford :-)

  We tend to recommend soemthing like an ultra-wide SCSI disk for best performance, but on smaller systems the IDE disks and newer ATA-66 type disks offer comparable performance for less money. Please feel free to let us know when we get out of date :-).

  NB: In general the seek speed is more important than the read speed.

  NB: If you are going to use a RAID system, then make sure if you have a big system, e.g. over 10,000 users, that you use hardware RAID rather than software RAID.

- **Disk Space per user**

  You can limit this to how much disk space you can afford with the settings, max_msgsize and the user_quota system.

  Typically on a system where the sysadmin promotes or enforces that users remove their mail from the POP server this works out to a very small, about 20kbytes per user. So if you work on about 40Kbytes per user that should be safe.

  Where you wish to allow users to leave mail on the POP server, they will of course need more space. A common limit for Hotmail type setups is 1MB per user, so 5 MB per user is probably considered quite generous. For inhouse systems you may want to limit it to something higher e.g. 10MB per user.

- **Non-user storage space**(e.g. log files, workarea, executables)

  About 25Mbytes. Things that will increase this,
  - the number of messages in DSMTP's queue and their sizes (work directory)
  - number of mailing lists and in particular the size of file and message archives for your lists (dlist\listname directories)
  - size and number of your log files, (log directory)

NWAuth with 100,000 users

Netwin's own external authentication module, NWAuth comes free with DMail, but how does it perform on big systems???

From our testing on a small Windows NT test machine (P133, 32Mbytes ram) with a user database of 100,000 users NWAuth performed exceptionally well.

nwauth.exe (the version distributed with dm25d):
Required 15Mbytes of memory.
User lookups were still instant.

Password checks were still instant.
Adding users or changing a password were still instant.

Products Downloads Prices Support Company

Security mail out Information Page

You can modify your EMAIL subscriptions for security notices on this page
http://netwinsite.com/cgi-bin/dnotice.cgi

FAQ on security issues

Recommended versions of CGI's

Robots running as root - **Security Note**

DMAIL Security Fault Notice 5 June 2000.

A fault was reported that allows root access to be gained.

We recommend all customers upgrade to

- DMail 2.7r (release) http://netwinsite.com/dmail/download.htm
- DMail 2.8k (beta)

If you cannot find an appropriate build for your operating system email support-dmail@netwinsite.com with details and we will provide a build as soon as possible.

Tehnicial Details

The fault is such that root access can be gained primarily to linux hosts using a web published exploit program.

On Linux to find out if your system has been attacked do this:
    grep "etrn" /usr/local/dmail/dwatch/*.ded
If that finds any lines dated within the last 3 days, then it is likely your system was attacked.

The fault could be exploited on other systems in future so it is essential that you upgrade even if you are not running linux.

We applogize for this fault we are not happy that it existed in the first place. We believe the above versions will be immune to all similar attacks. However we will also be investigating further in an attempt to find and remove any other possible exploits.

History of Security Fixes

see Known Bugs Page and Updates Page for further details.

- 2.8w and 2.9a, small but possible DList exploit fixed.

DMail (DSMTP, DPOP and DList) - Free Trial !!!

You may try out the DMail package free of charge for one month, simply by downloading the FULL version from our web site, as per the license agreement for this product.

It includes the DSMTP, DPOP and DList servers, management utilities and full documentation.

During this trial email support is provided free of charge, simply email support-dmail@netwinsite.com

If you decide to continue using DMail you then need to register your copy to receive a License key to keep it working. This ensures you do not waste the effort involved in configuring DSMTP, DPOP and DList for your situation. Once you receive your key you can continue using the DMail servers without any need to reinstall or modify settings.

Registration requests are normally sent by email and a key is emailed to you within two days of receipt of payment.

Download Full Version for Trial

The best way to register is to use our,

Secure Online Registration Form

Alternatively you can email or fax your registration to us.

To generate the registration details to email or fax to Netwin use the command

tellpop register (all licenses are for DMail, whether you wish to run just one or all three of the servers)

It will ask you a series of questions and then generates a register.txt file which you can

- Email to us at sales@netwinsite.com
- Fax to us at (+64) 9 630 0689
- Post to us, see Contacting Netwin (but we prefer email)

Contents:

A. Licensing and Support Guidelines

Evaluation:

DPOP, DSMTP and DList are products of Netwin Ltd. You may for the purpose of evaluating this product use the software at no charge for a single period, not exceeding 1 month following the month of installation..

After that period you MUST register your copy and pay the appropriate license fee or stop using it. If you are a University or School using this software for NON PROFIT purposes a substantial discount is available.

Current License and Support Options:

Netwin is committed to providing the best license and support options. When you purchase a DSMTP/DPOP/DList Server License, you'll receive these complimentary benefits: Prices.

- **Free technical support by email for 12 months.**
- **Free maintenance and update releases for the product or version purchased.**
- **Free any major new release of DSMTP, DPOP or DList for 12 months.**

Our expert support staff will provide you fast, focused support.

License life time:
Your DSMTP/DPOP/DList license allows you to continue using your version of the software forever, it does not expire with your support contract. However if you wish to run **new** versions released after the 1 year period then you must purchase a new license or take up a support and/or upgrade plan.

Porting

Netwin will probably port DSMTP/DPOP/DList to any UNIX'ish platform on request. We require a registration before performing this service as well as Telnet access to your machine.

Send Email to netwin@netwinsite.com if you have any questions.

B. How to register?

To register your copy

You must first download the free trial copy - you can't buy it without trying it

Execute the command:
tellpop register
and answer the registration questions.

This creates a file called REGISTER.TXT

Step 2:

Email the registration file, register.txt to netwin@netwinsite.com.

If you are registering as a State School or University please also confirm your request on headed notepaper by fax.

That's It:

You will receive an email reply (within 24 hours) confirming your registration or license or support contract depending on which options you selected. The confirmation will also include your **registration key** which will allow the software to continue to be used.

Payment Options

Payment is normally made by Credit Card. We can accept payment via Visa, Master Card and American Express. Simply enter the name as it is specified on the card, the card number and expire date into the Registration Form. This data is encrypted for protection. However if you prefer you can FAX your details to us on +64 9 6300 689.

If you will paying by cheque or bank transfer please note this in the comments section of the registration form when you register. When we have received your funds (via Credit Card, cheque or bank transfer) we will email you your full key to be followed by an invoice (by post).

Where you are unable to pay by Credit Card we will always email you a temporary key on receipt of your registration form to cover the time it takes to physically receive your payment.

Cheques should be made payable to "Netwin LTD" and posted to; Netwin LTD, PO Box 27574, Mount Roskill,
Auckland, New Zealand. (Use this Currency Converter to calculate the amount owed in your own currency for
cheques. Then use the back button on your browser to return to this page) Prices:

A bank transfer may be the most economical method of payment. Please email netwin@netwinsite.com for our Bank Account details.

C. End User License Agreement For DMail Mail Servers

IMPORTANT - Read this License Agreement Carefully:

This License Agreement (LA) is a legal agreement between you (either an individual or a single entity) and Netwin Limited for the Netwin Software Product identified above, which includes computer software, and may include associated media, authorization keys and online or other documentation ("Software Product" or "Software"). By installing, copying, or otherwise using the Software Product, you agree to be bound by the terms of this LA. If you do not agree to the terms of this LA, you may not install, copy or otherwise use the above software.

Software Product License

The Software Product is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Software Product is licensed, not sold.

1 Grant of License. This License Agreement grants you the following rights:

1.1 Systems Software. You may install and use one copy of the Software Product on a single computer, which may be connected at any point in time to an unlimited number of computers operating on one or more networks.

1.2 License FEES. You may use the software for a single period not exceeding 1 calendar month following the month of installation for the purpose of evaluating the software (the evaluation period) at no charge. To use the Software beyond the evaluation period you MUST register your copy and pay the applicable fee(s).

2. Description of other Rights and Limitations

2.1 Limitations on Reverse Engineering, Decompilation, and Disassembly. You may not reverse engineer, decompile, or disassemble the Software Product, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

2.2 Rental. You may not lease or rent the Software Product.

2.3 Proprietary notices. You may not remove any proprietary notices or labels on the Software Product.

2.4 Software Transfer. You may permanently transfer all of your rights under this LA, provided you advise Netwin LTD, you retain no copies, you transfer all of the Software Product (including all component parts, the media and printed materials and any upgrades), AND the recipient agrees to the terms of this LA. If the Software Product is an upgrade, any transfer must include all prior versions of the Software Product.

2.5 Termination. You may terminate this Software Product License at any time. In addition, without prejudice to any other rights, this Agreement and the license granted hereunder will terminate automatically if you fail to comply with the terms and conditions described herein. Upon termination, you must destroy all copies of the Software and Documentation. Your obligations to pay accrued charges and fees shall survive any termination of this Agreement.

2.6 Authorization Keys. Authorization keys may be installed and enabled for use in only one license control utility. You may not modify or make inoperable authorization keys or license control utilities.

3. Copyright

All title and copyrights in and to the Software Product, and accompanying printed materials are owned by Netwin LTD. The Software Product is protected by copyright laws and International treaty provisions. Therefore you must treat the software product like any other copyrighted material except that you may either (a) make one copy of the SOFTWARE PRODUCT solely for backup or archival purposes, or (b) install the software product on a single computer provided you keep the original solely for backup or archival purposes. The DMail software DSMTP, DPOP and DList is Copyright © 1995 Netwin LTD, New Zealand, All rights reserved. The software remains the sole and exclusive property of Netwin at all times.

4. Updates and Technical Support:

Upon registration and payment of appropriate fee(s) Netwin LTD for the specified period from the date of registration of the Licensed Software will make available to you Technical Support in the manner and under the guidelines set forth in the Licensed Software User Documentation, which may be modified from time to time by Netwin at its discretion without notice. Netwin may, from time to time, revise or update the licensed software. In so doing, Netwin incurs no obligation to furnish such revision or updates to you. Updates and further support terms are available to you on the same basis as Netwin makes them available to its other licensees at then current prices.

5. Disclaimer of Warranty

The Software is provided on an "AS IS" basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and non-infringement. The entire risk as to the quality and performance of the Software is borne by you. Should the Software prove defective, you and not Netwin or its suppliers assume the entire cost of any service and repair. In addition, the security mechanisms implemented by Netwin software have inherent limitations, and you must determine that the Software sufficiently meets

your requirements. This disclaimer of warranty constitutes an essential part of the agreement.

6 Limitation of Liability

To the maximum extent permitted by applicable law, any conditions or warranties imposed or implied by law are hereby excluded. Consumers may however have the benefit of certain rights or remedies in respect of which liability may not be excluded. Insofar as such liability may not be excluded then to the maximum extent permitted by law, such liability is limited, at Netwin's exclusive option, to either (a) the price paid for the SOFTWARE or (b)replacement of the SOFTWARE.

7 Exclusion of Liability/Damages

The following is without prejudice to any rights you may have at law which cannot legally be excluded or restricted. You acknowledge that no promise, representation, warranty or undertaking has been made or given by Netwin (or related company) to any person or company on its behalf in relation to the profitability of or any other consequences or benefits to be obtained from the delivery or use of the SOFTWARE and any accompanying Netwin Supplied software, manuals or other materials. You have relied upon your own skill and judgment in deciding to acquire the SOFTWARE and any accompanying manuals and other materials for use by you. Except as and to the extent provided in this agreement, neither Netwin nor any related company will in any circumstances be liable for any other damages whatsoever (including, without limitation, damages for loss of business, business interruption, loss of business information or other indirect or consequential loss) arising out of the use, or inability to use, or supply or non-supply, of the software and any accompanying written materials. Netwin's total liability under any provision of this agreement is in any case limited to the amount actually paid by you for the software.

Netwin will make reasonable efforts to solve any reported problems but we must limit our legal liability for obvious reasons to an extent which is proportionate to the commercial value of this transaction.

DMail Utilities Download Page

This is a slowly evolving page :-)

Netwin and some of our customers have produced add-ons for DMail. This page provides links to those, along with any information available, on the utility that is not provided with the utility.

All those written or owned by netwin are marked; 'Netwin'. All others have been written by third parties and as such you should use them at your own risk - please do not assume that they have been tested (although most have) by Netwin, or that Netwin proclaims them safe to use or even functional with its products :-)

ODBCAuth - External Authentication

An External Authentication module written by us for talking to an ODBC driver that connects to a User Database on a Windows box.

Designed with the following Databases in mind,
Microsoft Access, Microsoft SQL Server, ORACLE.

For details see the odbcauth page in the dmail manual,
ODBCAuth.htm

NB: This is currently a BETA authentication module. Contact,
DMail Support if you have any problems with it.

Latest download is,
ftp://ftp.netwinsite.com/pub/dmail/beta/odbcauth10d.zip

Known Problems:

NB: ODBCAuth only supports the BASIC External Authentication Protocol, i.e. user checks and lookups - it does not allow searches and adding users via it.

NB: ODBCAuth with MS SQL Server: In version 1.0d we have fixed the bug with MS SQL 7 from version 1.0c.

History:

1.0d 21 Jun 2000 KB + TRW
- fixed bug with SQL 7
- removed 'debug' ini setting so cannot accidently run in debug mode.

1.0c 8 May 2000 TRW
- all lookups are done with lowercase usernames when lowercase_username true in ini file.
- ensures that username returned matches that given with orig_user variable.
- added setting, log true, which causes debug logging but nothing extra printed to screen.

1.0b - 27 April 2000 TRW
- fix log_path setting, so when set don't get small extra log in exe directory.
- remove printfs so that 'type 1' and 'type 5' etc are not returned in non-debug mode
- change database connection errors to return - DEAD instead of -ERR so that users get told to come back later.
- made -DEAD messages return nice, 'Database Problem, please try again later' message.
- make log lines not have blank line between them (imsg etc print with \n insted of NEXT_LINE).

1.0a - 22 Mar 2000 TRW put on the site.

DRespond

- Netwin's latest version of the autresponder. Run drespond or drespond.exe from a command line to see useage information. You should set up a forward rule or an alias to the drespond robot for the user whose mail you wish to have responded to. See robots for more information.

'Netwin' dres1602.zip - for NT
NB: This file is included in the DMail 2.4h, and above, distribution set for all platforms.

Columns - for creating batch files to add users

- A small utility written by Netwin, which can be used to create a batch file that adds a list of users to an External Authentication Module, e.g. NWAuth

Run cols(.exe) without any arguments to get the following useage information:

Columns Version 1.0a

Useage:
Columns allows you take data from specific columns
in each line of an input file and write it out to a
file (cols.out) in a specified format.

cols filename <-try> -delim ";" -out "nwauth -set $4 $5"

Where $4 indicates the 4th column.
Columns are delimeted by the single character
given with the -delim option.
Use the -try option to test your parameters (no
output file will be created).

For example:

This example shows four users being added to the nwauth external authentication routine, Note, in order to do this you do need to know the user's passwords in an unencrypted form.
The command line,

cols input.txt -delim ";" -out "nwauth -set $3 $4"

will turn the following input file,input.txt,

```
junk;junk;bob;bobs_password;junk;junk;;;
junk;junk;fred;freds_password;junk;junk;;;
junk;junk;rupert;ruperts_password;junk;junk;;;
junk;junk;joe;joes_password;junk;junk;;;
```

into the output file, cols.out,

```
nwauth -set bob bobs_password
nwauth -set fred freds_password
nwauth -set rupert ruperts_password
nwauth -set joe joes_password
```

This file can then be renamed, cols.bat and run by simply entering cols at the command prompt.

Additional Use: To add a prefix/suffix to all usernames

The flexibility of the output string allows you to use Columns to add suffixes and prefixes to the usernames.

E.g. To add a domain suffix @netwinsite.com onto the end of all usernames in your nwauth.txt file,

```
tam:lcYQf:fwd="" info="" groups=""
bob:mlUW[:fwd="" info="" groups=""
trev:qwEWJY:fwd="" info="" groups=""
fred:eqO_FA:fwd="" info="" groups=""
fred99:dkM]GUPg:fwd="" info="" groups=""
fred9:xiaIeUF:fwd="" info="" groups=""
tam99:fsI_`Od:
tam55:nyQ]JEN:
```

run cols on that file with the following command line,

cols nwauth.txt -delim ":" -out "$1@netwinsite.com: $2:$3"

This gives the following output file, cols.out,

```
tam@netwinsite.com:lcYQf:fwd="" info="" groups=""
bob@netwinsite.com:mlUW[:fwd="" info="" groups=""
trev@netwinsite.com:qwEWJY:fwd="" info="" groups=""
fred@netwinsite.com:eqO_FA:fwd="" info="" groups=""
fred99@netwinsite.com:dkM]GUPg:fwd="" info="" groups=""
fred9@netwinsite.com:xiaIeUF:fwd="" info="" groups=""
tam99@netwinsite.com:fsI_`Od:
tam55@netwinsite.com:nyQ]JEN:
```

You can then replace your nwauth.txt file with this file renamed as nwauth.txt. NOTE: To replace nwauth.txt you must make sure that there are no entries in nwauth.add and that all instances of nwauth are stopped (i.e. shutdown all servers and check process list for any stray instances).

Download:'Netwin'
Download for NT

IMAPD - IMAP Server

IMAPD is a IMAP server to run alongside or instead of DPOP, to allow users to connect to an IMAP server to read their mail, instead of connecting to the POP server.

Notes:

1. The IMAP server supplied here is based upon Mark Crispin's IMAP4rev1 server (RFC 2060), version 4.4, which has been modified to run in conjunction with DMail. Mark Crispin's source can be obtained from ftp://ftp.cac.washington.edu/mail/imap.tar.Z See Imap Release Notes for his copyright notice and release notes.
2. We supply this port 'as is where is'. DMail Support staff can offer limited help with problems with the unpacking of the distribution set and/or problems with our port.

3. We recommend you run version 2.5g or higher of DMail.
4. Windows NT users must download a new version of dwatch.exe so that they can trial IMAPD until DMail 2.5h is available.
   (This replaces the existing dwatch.exe. eg. c: \dmail\dwatch\dwatch.exe. Remember to stop the dwatch service first)

For installation instructions see,
IMAPD Server

For configuration instructions see,
IMAPD Configuration

Download: 'Netwin'

IMAPD 4.4.3t - Windows NT (422 kbytes)

IMAPD 4.4.3u - Linux (782 kbytes)

IMAPD 4.4.3u - Linux libc6 (449 kbytes, linux_libc6 - for RH Linux 5.2 etc.)

IMAPD 4.4.3u - FreeBSD 3 (748 kbytes)

IMAPD 4.4.3u - Digital Unix(OSF) (875 kbytes)

IMAPD 4.4.3u - Solaris(Sparc) (864 kbytes)

IMAPD 4.4.3u - Solaris(x86) (1186 kbytes)

NWAuth

- Netwin's latest version of the nwauth external authentication module. See external authentication for more information.

'Netwin'
NB: This file is included in all distribution sets, we recommend that you use the one from the latest beta release. Please note that if you are using WAdduser, then you must upgrade it to the same version as the nwauth you are using.

LDAPAuth - External Authentication Module

- Netwin's latest version of the LDAP external authentication module. See the external authentication section of the dmail manual for more general information and the LDAPAuth page for specific information.

NB: LDAPAuth is now built with the DMail distribution set, so check for the current version in your distribution set first. If not there check the list below. If you can't find the version you want for your OS then please email DMail Support.

'Netwin'
Version 1.0c: Windows Platforms:
ldp10c.exe
Version 1.0c: UNIX Platforms:
ldp10c_linux.tar.Z
ldp10e_libc6 (binary only)
ldp10c_solarissparc.tar.Z
ldp10i_solarissparc.tar.Z

Please email DMail Support if you would like us to try building LDAPAuth on another platform, or you can download the source with the link below,
ldp10c_linux.tar.Z

Known bugs in versions:

- 1.0c fwd field returned with incorrect syntax, i.e. fwd=blah rather than fwd="blah"
- 1.0c can core dump on some systems when given a null username

Recent History:

- 1.0i trw added -version option.
- 1.0h chrisp - Added setting ldap_scope, defaults to LDAP_SCOPE_SUBTREE can also be set to LDAP_SCOPE_ONELEVEL, LDAP_SCOPE_BASE
- 1.0g chrisp - added user_extend true, if defined then if username has ldap search fields added to it they are added to the base search (no they replace the base and they must be added to the username, e.g. chrisp,ou=developers,o=netwin
- 1.0f Chrisp added setting 'ldap_search_name' which defaults to 'mail' and also changed so if domain is blank then domain name is not added.
- 1.0e (adding the stuff from 1.0d again)
  28 July 2000 TRW
  - fix fwd="" stuff.

DNAuth - beta

- Netwin's latest version of the dnauth external authentication module, which reads from DNews's users.dat file.
Click here for more information.

'Netwin'
Version 1.0a is not considered worth using, 1.0b is :-)

NB: This file is included in the DMail 2.5d, and above, distribution set for all platforms.

Keith Steven's NT ODBC Authenticator

An ODBC external authentication module for DMail on Windows NT.

Follow this link for more information . . .
ODBC Authenticator
(if you need the source for this authenticator then note that there is a small charge)

(Netwin) FixHash - for changing between hashing methods

NB: This is a new Netwin utility which is still in its beta form. We have tested it on both NT and Linux, and checked that it does not alter directory or file permissions, but we have not tested it on a large system as yet. Please let us know if you try it and how you get on.

This utility will move all directories and files in a mail spool from one hashing system to another. The systems that it works with are those defined under the hash_spool dmail.conf setting,
i.e. hash_spool 0 = no hashing,
hash_spool 1 = 1 extra directory level,
hash_spool 2 = 2 extra directory levels.

Here is an example of running this program to change from hash_spool 0 to hash_spool 2, where at present all mail is in drop files,
/var/mail/user1
/var/mail/user2
/var/mail/user3
etc.

./fixhash -hash 2 -path /var/mail

Notes:

1. You must add the option, -doit for fixhash to actually do anything. If you don't it simply writes to screen what it would do.
2. Enter just ./fixhash, (or simply fixhash on NT) to get further useage information
3. Remember to set your hash_spool setting in dmail.conf so that DSMTP and DPOP know where to find the drop and bin files
4. You MUST shutdown DSMTP, DPOP and DList before running fixhash
5. Fixhash moves DPOP's bin_files for every user if it finds them in the same directory as the user's drop file.

NB: Fixhash will be in all distribution sets from 2.8r onwards, so you should have it on your machine

already!

Download BETA for NT:
ftp://ftp.netwinsite.com/pub/dmail/beta/fixhash10a.zip

Download BETA for Linux:
ftp://ftp.netwinsite.com/pub/dmail/beta/fixhash10a.Z

Download BETA for AIX:
ftp://ftp.netwinsite.com/pub/dmail/beta/fixhash10a_aix.Z

Download Source and UNIX Makefile:
ftp://ftp.netwinsite.com/pub/dmail/beta/fh10asrc.tar.Z

Products Downloads Prices Support Company

UNIXAuth - External Authentication Module for UNIX password file

NB: This authentication module is currently in beta form only!

UNIXAuth is an external authentication program, it uses Netwin's External Authentication Protocol. Documentation on this standard can be found in the User Administration section of this DMail Manual.

UNIXAuth accesses and writes to the UNIX operating system user database, e.g. i.e. the /etc/passwd file or shadow password file.

UNIXAuth has been designed to replace the 'unix_user' option of the authent_method dmail.conf setting. It allows DMail to run an external authentication module but still access /etc/password. It also allows our web admin tool, NetAuth to be used to administer users in the /etc/password file.

UNIXAuth uses the standard operating system user authentication calls (as the DMail servers do when authent_method is set to unix_user) to access the /etc/password file so that it works with any standard authentication system like shadow password files.

On this page ...

---

1. Ini settings

Put the unixauth.ini file in the same directory as the unixauth executable, e.g. /usr/local/dmail. You can use any of the following settings...

path <path>	Path sets the path for locating the ini file, and also determines where the log file will be created. For example running unixauth -path /var/log will cause it to load the unixauth.ini file from the /var/log directory if in the /var/log/unixauth.ini file there is a path setting unixauth will then create it's log file in the new directory.

log <level>	This setting can be used as just "log" or you can specify a level of logging like so, "log debug" or "-log debug". It has three valid logging levels error, info, and debug.
debug	This setting causes unixauth to run with logging set to debug mode, it is equivalent to a "log debug" or "-log debug".
group <name>	This specifies the group that user created will belong to, this group must belong on the host the users are being added to. This can be a string or on Unix a GID.
script_path	Path to the shell files, any shell setting must exist in this path, there is no default for this, if no setting is found shell settings are assumed to be from the root.
uname	Set default "name" field, automatically set to "Unix Auth <version> User".
shell	Default "shell" field. Automatically set to "/bin/bash".
base_uid	Base number to start adding new UID's. Automatically set to "100".
home_path	Default base-directory for "home" field. Actual "home" field is set to <base>/<username>. e.g. /home/root. Automatically set to "/home"

EXAMPLE unix ini file <unixauth.ini>

```
path /usr/local/dmail
log error
group users
base_uid 500
```

---

2. Error codes

These errors were written in an attempt to be descriptive enough so that you could problem solve without too much hassle. If you are completely stumped and have no idea why you are recieving an error then it could be our fault entirely :-), so simply email the error and what you were trying to do to "Sysauth Help" <support-netauth@netwinsite.com>.

"-ERR ##:Unknown error has occurred."
There was an error. We need the error number ## to determine what went wrong.

"-ERR Not a valid command (nocommand) use help"
You didn't enter a command.

"-ERR Not a valid command (<command>) use "help""
You entered <command> which was not recognised by unixauth as a valid command.

"-DEAD Unable to open {<file>,<reason>}"
"-DEAD Error with open of password file {<file>,<reason>}"
"-DEAD Unable to open password file {<file>,<reason>}"
"-DEAD Error with open of group file {<file>,<reason>}"
"-DEAD Unable to open temporary file {<file>,<reason>}"
"-DEAD Error with lock of password file {<file>,<reason>}."
"-DEAD Error with un-lock of password file {<file>, <reason>}."
"-DEAD Unable to create init.log, <reason>"
Sysauth couldn't open / close the required file <file> for <reason>.

"-DEAD Error with lock of password file {<file>,<reason>}."
Unable to obtain a lock for the <file>, for <reason>.

"-DEAD Unable to close log file {<reason>}"
Sysauth could not close it's log file for .

"-ERR Error writing current entries {<reason>}."
Sysauth encountered an error while trying to copy entries in the password and/or group file.

"-ERR Error removing current password file, <reason>"
Sysauth could not remove old copy of password file.

"-ERR Error copying temporary file -> password file, <reason>"
Sysauth could not replace the password file with an updated copy.

"-DEAD No more UID's available."
Sysauth could not locate a free UID.

"-ERR Invalid user info parameter."
Either the home directory, name and / or shell parameter is in error.

"-ERR Unknown error occurred."
Something strange has happened.

"-ERR Home directory must exist."
The home directory given must exist.

"-ERR setting <setting> incorrect"
The format of setting <setting> was incorrect, user setting="value".

"-ERR Shell must exist."
The shell file given must exist.

"-ERR Unknown user info {}"
The <setting> was an unknown user setting.

"-ERR Error locating GID for {}, group does not exist."

The group <group> could not be found.

---

3. Output

The reply messages are part of the Netwin standard External Authentication Protocol.

Command	Message
set	+OK User <name> added to the database
del	+OK Deleted user successfully
lookup	+OK <user> config 0 <info>
	-ERR <user> not found
check	+OK <user> config 0 <info>
	-ERR <user> password wrong or not a valid user
search	+DATA ...
	+DATA ...
	+OK search complete <number> items found
version	+OK NT Auth version <version number>
help	+DATA Valid commands
	+DATA <command>
	+DATA <command>
	+DATA <command>
	+OK

---

4. Download and History

Generally you will find the latest download in with your distribution set. You can check what version of unixauth you have by running unixauth at a command prompt, e.g.,
c:\dmail\unixauth -version

If we put a download on the site for a version, it will listed in the history below as a link.

History:

1.0a 14 Aug 2000 (also in dmail 2.8 versions)
unixauth28s_libc6

NTAuth - External Authentication Module for Windows User Database

NB: This authentication module is currently in beta form only!

NTAuth is an external authentication program, it uses Netwin's External Authentication Protocol. Documentation on this standard can be found in the User Administration section of this DMail Manual.

NTAuth accesses the Windows operating system user database, i.e. the OS's User Accounts which you would normally administer with something like, User Manager.

NTAuth has been designed to replace the 'nt_user' option of the authent_method dmail.conf setting. It allows DMail to run an external authentication module but still access Window's user accounts. It also allows our web admin tool, NetAuth to be used to administer Window's user accounts.

NTAuth has been designed to work with Windows NT and Windows 2000.

On this page ...

---

1. Ini settings

Put the ntauth.ini file in the same directory as the ntauth executable, e.g. c:\dmail\. You can use any of the following settings...

path <path>	Path sets the path for locating the ini file, and also determines where the log file will be created. For example running ntauth -path \here will cause it to load the ntauth.ini file from the \here directory if in the \here\ntauth.ini file there is a path setting ntauth will then create it's log file in the new directory.
log <level>	This setting can be used as just "log" or you can specify a level of logging like so, "log debug" or "-log debug". It has three valid logging levels error, info, and debug.
debug	This setting causes ntauth to run with logging set to debug mode, it is equivalent to a "log debug" or "-log debug".

group <name>	This specifies the group that user created will belong to, this group must belong on the host the users are being added to. This can be a string or on Unix a GID.
script_path	Path to the script files, any script setting must exist in this path, there is no default for this, if no setting is found script settings are assumed to be from the root.
host <name>	This specifies the host users are added, deleted and verified on. This is an NT only setting.

EXAMPLE NT ini file <ntauth.ini>

```
path c:\dmail
log error
host internet.mail.com
group Guests
```

---

2. Error codes

These errors were written in an attempt to be descriptive enough so that you could problem solve without too much hassle. If you are completely stumped and have no idea why you are recieving an error then it could be our fault entirely :-), so simply email the error and what you were trying to do to "Sysauth Help" <support-netauth@netwinsite.com>.

"-ERR ##:Unknown error has occurred."
There was an error. We need the error number ## to determine what went wrong.

"-ERR Not a valid command (nocommand) use help"
You didn't enter a command.

"-ERR Not a valid command (<command>) use "help""
You entered <command> which was not recognised by ntauth as a valid command.

"-DEAD Unable to open {<file>,<reason>}"
Sysauth couldn't open a required file <file> for <reason>.

"-DEAD Unable to close log file {<reason>}"
Sysauth could not close it's log file for <reason>.

"-ERR Group %s does not exist."
The group specified to add users to didn't exist.

"-ERR User was already a member of group %s."

The user already belongs to the group specified.

"-ERR ##:Unknown group error."
There was an error adding the new user to a group. We need the error number ## to determine the error.

"-ERR The user does not have access to the requested information."
"-ERR The operation is allowed only on the primary domain controller of the domain."
"-ERR This operation is not allowed while you are a member or your current group."
"-ERR This operation is not allowed on the last administrative account."
"-ERR Incorrect Privilege (Dmail setup may be incorrect)."
The privilege required to carryout the command is not held by this user.

"-ERR The computer name is invalid."
The computer specified by the "host" setting can not be found.

"-ERR The group already exists."
A group by that name exists.

"-ERR The password is shorter than required."
The password supplied is too short.

"-ERR The password is invalid."
The password is incorrect.

"-ERR User account not found."
The user account does not exist.

"-ERR Home directory must contain a drive letter."
Home directory has to contain a drive letter specifying the drive on which it exists.

"-ERR Home directory must exist."
Home directory must exist.

"-ERR Script must exist."
The script specified does not exist.

"-ERR Setting %s incorrect"
Format for the setting was incorrect. Use setting="value".

"-ERR Parameter ## is in error."
The user information parameter ## is incorrect see below..

If you recieve any of the following errors then we have probably fouled something up, please email us the error at "Sysauth Help" <support-netauth@netwinsite.com>

1. "-ERR Username invalid."
2. "-ERR Password invalid."
3. "-ERR Password age invalid."
4. "-ERR Privilage invalid."
5. "-ERR Home directory invalid."

6. "-ERR Comment invlaid."
7. "-ERR Flags invalid."
8. "-ERR Script path invalid."

If you recieve any of the following UNICODE errors then again we have made a mess of something, please email us the error at "Sysauth Help" <support-netauth@netwinsite.com>

"-ERR Unable to convert to unicode, insufficient buffer space."
"-ERR Unable to convert to unicode, invalid flags."
"-ERR Unable to convert to unicode, invalid parameter."
"-ERR Unable to convert to unicode, no possible translation."
"-ERR ##:Unable to convert to unicode."

---

3. Output

The reply messages are part of the Netwin standard External Authentication Protocol.

Command	Message
set	+OK User <name> added to the database
del	+OK Deleted user successfully
lookup	+OK <user> config 0 <info>
	-ERR <user> password wrong or not a valid user
check	+OK <user> config 0 <info>
search	+DATA ...
	+DATA ...
	+OK search complete <number> items found
version	+OK NT Auth version <version number>
help	+DATA Valid commands
	+DATA <command>
	+DATA <command>
	+DATA <command>
	+OK

MySQLAuth External Authentication Module for DMail

An external authentication module that allows DMail to do user lookups on a MySQL Database.

Installing MySQLAuth

When you look for MySQLAuth in your DMail distribution, you should find two files:

unix	nt
mysqlauth mysqlauth.ini	mysqlauth.exe mysqlauth.ini

(If you cannot find these files see, Download and History for MySQLAuth)

Place these files into your dmail directory and update dmail.conf to point to this external module.

e.g. on UNIX
authent_method external
authent_process /usr/local/dmail/mysqlauth
and on Windows:
authent_method external
authent_process c:\dmail\mysqlauth.exe

Then you must edit the mysqlauth.ini file so that it points to your MySQL server. More details on this step are provided on the rest of this page.

NB: You should locate the ini file in the same directory as the mysqlauth binary or as,
/etc/mysqlauth.ini (on UNIX based platforms)
or,
c:\winnt\system32\mysqlauth.ini (on Windows platforms)

And then RESTART both DSMTP and DPOP.

NB: you must RESTART both DSMTP and DPOP when ever you change your mysqlauth.ini file, as they spawn copies of MySQLAuth which only reads mysqlauth.ini at startup.

And that is it. For details of other External Authentication Settings see the User Administration section of the DMail Manual.

Creating/Using a MySQL Database

MySQLAuth requires a MySQL database which is setup and working. The database that is setup must have a username and a password that is encrypted using the mysql command PASSWORD(). You can either create a new database/table for MySQLAuth or use a current database that has usernames and passwords.

New DataBase:

Below are instructions on how to setup a brand new database and table to work with MySQLAuth, with all of the features that MySQLAuth provides.

```
CREATE DATABASE maildb

USE maildb;

CREATE TABLE maildb (
      username CHAR(128) binary DEFAULT '' NOT NULL,
      passwd CHAR(128) DEFAULT '*' NOT NULL,
      forward CHAR(255) DEFAULT '',
      quota CHAR(20) DEFAULT '',
      mailmask CHAR(18) DEFAULT '0.0.0.0' NOT NULL,
      maildrop CHAR(255),
      PRIMARY KEY (username)
);
```

To insert a brand new account into this account using your sql the command is:

```
INSERT INTO maildb VALUES
    ('test@test.org', PASSWORD('test'), '', '100k',
     '0.0.0.0','/usr/spool/mail/test');
```

This will create a user called test@test.org with the username of test.

To manually remove a user the command is:

```
DELETE FROM maildb WHERE username='test@test.org'
```

The inserting and deleting of users is taken care of by MySQLAuth, the above is to show how you would manually do this.

The ini settings for the above would look like the following:

```
mysql_server your.sql.server
mysql_login login
mysql_password password
```

domain your.default.domain

mysql_mail_user_db maildb
mysql_mail_user_table maildb

field_username username
field_password passwd
field_forward forward
field_quota quota
field_mailmask mailmask
field_maildrop maildrop

Using Current DataBase:

You already have an existing database that you wish to use then as long as the usernames are unique and the password field is encrypted using the PASSWORD() mysql command, then you should be able to just change the ini settings to point to this database, table and field label names.

eg. If you have a database called '**accounts**' and a table called '**mail_users**' that stores all sorts of information but has the username field names '**name**' and the password field called '**pwd**' then the ini settings that you required are:

mysql_server your.sql.server
mysql_login login
mysql_password password

domain your.default.domain

mysql_mail_user_db accounts
mysql_mail_user_table mail_users

field_username name
field_password pwd

If you have the ability to also store the mail quota or forwarding then you can add these ini settings as well.

ie. field_forward forward
field_quota quota

Check Install

Once MySQLAuth is installed it is best to then check that MySQLAuth is working correctly. You will need to run MySQLAuth from the command line (in a dos box on Windows) and try the following.

NOTE: C: - Client, S: - Server

```
./mysqlauth

C: set test_account password
S: +OK 'test_account@test.org' has been added to database

C: set test_fwd password fwd="test_account@domain1"
S: +OK 'test_account@test.org' has been added to database

C: set test_quota password quota="100k"
S: +OK 'test_account@test.org' has been added to database

C: set test password fwd="test_account@domain1" quota="100k"
S: +OK 'test_account@test.org' has been added to database

C: search *
S: +DATA test_quota@test.org config 0 quota="100k"
S: +DATA test_fwd@test.org config 0 fwd="test_account@domain1"
S: +DATA test_account@test.org config
S: +DATA test@test.org config 0 fwd="test_account@domain1"
quota="100k"
S: +OK Search Completed 4 items found

C: lookup test_quota
S: +OK test_quota@test.org config 0 quota="100k"

C: lookup test
S: +DATA test@test.org config 0 fwd="test_account@domain1"
quota="100k"

C: check test password
S: +OK test@test.org config 0 fwd="test_account@domain1" quota="100k"

C: check test incorrect
S: -ERR test@test.org password wrong or not a valid user

C: del test
S: +OK 'test@test.org' has been deleted

C: del test_account
S: +OK 'test_account@test.org' has been deleted

C: del test_quota
S: +OK 'test_quota@test.org' has been deleted

C: del test_fwd
S: +OK 'test_fwd@test.org' has been deleted
```

The acutal response with quota and fwd might vary if the mysql database does not have quotas or fwd setup. To display the complete list of commands that MySQLAuth supports enter the command:

```
help
```

Information about the protocol being used can be found in the DMail manual:

User Administration , External Authentication Protocol section

---

MySQLAuth.ini settings

If you need an mysqlauth.ini to start with, download, mysqlauth.ini.

Label	Example	Default	Explanation
debug	true	false	Enables verbose debug output.
domain	your.default.domain	none	This is default domain that is appended to any username that does not already have a domain setting attached.
field_username	user	username	The label of the username field in your table.
field_password	password	passwd	The label of the password field in your table.
field_forward	fwd	forward	The label of the forward field in your table. This is the fowarding information that DMail uses to determine if mail for the account looked up should be delivered to a different address.
field_quota	quota	none	The label of the quota field in your table. This is the disk quota the user has. e.g. 100k, 10M
field_mailmask	mailmask	none	The label of the mailmask field in your table. This is the IP mask that MySQLAuth checks against the from IP of the user. If specified, this forces the user to connect only to the specified IP in order to collect mail (POP).
field_maildrop	maildrop	none	The label of the maildrop field in your table. This is the location where dmail will drop the user mail files. NB: if the field value is empty in the database (NULL) mysqlauth will return the keywork 'config' indicating that the server knows where to locate the drop file.
log_path	mysqlauth.log	auth.log	This is the mysqlauth log file.
mysql_server	your.sql.server	none	This is the IP or name of the computer hosting the mysql server.
mysql_login	username	none	This is the username that has access to the correct database and table that stores the usernames and passwords.
mysql_password	password	none	This is the password that is required and used in conjuction with mysql_login.

mysql_mail_user_db	user_data	maildb	This is the database name where the mail usernames are stored.
mysql_mail_user_table	users_list	maildb	This is the table name that is within the mail user database that has the user details.
mysql_mail_uid	99	0	This is the unique mail ID.NB: by default mysqlauth returns 0 for this which indicates to the DMail Servers that the userid should not be checked.

A step by step example of creating a mysqldatabase on NT

Below is an unedited example of setting up a mysql database to test on on a Windows NT box.

1. Downloaded from a mysql mirror site (they like it when you use a mirror),
mysql-shareware-3.22.32.win.zip
and unpacked to,
d:\mysql

2. Edited example file, d:\mysql\my-example.cnf that came with it, so that, all c:'s were changed to d:, and saved as,
c:\my.cnf
(notepad named it c:\my.cnf.txt for me so I did a rename on it in a dos prompt)

3. To install as service on NT,

d:\mysql\bin>mysqld-shareware --install

NB: when I then tried to start service in ControlPanel|Services, it failed with a message saying that the process had stopped itself.

so entered,

d:\mysql\bin>mysqld-shareware --debug
and at the top of all the mess was an error about a setting in the my.cnf file that it did not like.

NB: if I ran it from a command line, it started and could not be stopped even with a CTRL-C - I had to kill it in process list of task manager. In the bin dir was an exe mysqlshutdown, which just popped up a window (Icon in system tray), but I could not get it to stop the process - maybe it is for stopping the service? - no does not seem to affect that either - kind of cute icon though :-)
The dos command,
net stop mysql
does work as does,
net start mysql
.

4. Creating root user.

NB: our sqlauth module makes you send a password. By default the mysql daemon has a root user with no password. So you need to set the root password to something.

From the mysql manual ...
(The following example starts by removing the anonymous user, that allows anyone to access the 'test' database)

d:\mysql\bin\mysql mysql
mysql> DELETE FROM user WHERE Host='localhost' AND User='';
mysql> QUIT
d:\mysql\bin\mysqladmin reload
d:\mysql\bin\mysqladmin -u root password your_password

NB: I had problems getting the root password set after deleting it. I ended up entering the following to set the root password.

D:\mysql\bin>mysqlc -u root mysql
Reading table information for completion of table and column names

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 16 to server version: 3.22.32-shareware-debug

Type 'help' for help.

mysql> UPDATE user SET Password=PASSWORD('qwerty') WHERE user='root';
Query OK, 2 rows affected (0.02 sec)
Rows matched: 2 Changed: 2 Warnings: 0

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.03 sec)

Now in c:\my.cnf edit client section and add,
password=qwerty
and save.

Now you should be able to connect with the client program as user root with the new password...

5. Now follow the examples in the manual.

In section 8.4 of the manual file,
d:\mysql\docs\manual.htm
start client (on local machine - same as ...

D:\mysql\bin>mysqlc -u root -pqwerty

(can also do,
D:\mysql\bin>mysqlc -u root -p
and it prompts for password so that your command prompt buffer does not contain your password if you are worried about that sort of security)

(on non-local machine,

D:\mysql\bin>mysqlc -u root -pqwerty -host=1.2.3.4
)

(NB: you can also include a database name on the end of the line to connect to a specific database, e.g. to connect to a database called, 'mydb' enter,
D:\mysql\bin>mysqlc -u root -pqwerty mydb
)

everytime you see 'mysql>' from now on it indicates that you are at the prompt in the mysqlc client program...

6. Create test database and a table within it ...

mysql> CREATE DATABASE maildb;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE TABLE maildb (username VARCHAR(20),password VARCHAR(20),forward VARCHAR(20));
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW TABLES;

```
+------------------+
| Tables in maildb |
+------------------+
| maildb           |
+------------------+
1 row in set (0.01 sec)

mysql> DESCRIBE maildb;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(20) | YES  |     | NULL    |       |
| password | varchar(20) | YES  |     | NULL    |       |
| forward  | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

7. Loading users in from a text file ...

I then made a file d:\mysql\users.txt which looked like,

tam pass \N
test test \N

where \N stands for NULL meaning no entry for that field. and then tried to load it. You'll see I assumed a few things about file paths incorrectly. I remember reading in the manual somewhere about using / instead of \ (or using \\).

```
mysql> LOAD DATA LOCAL INFILE "users.txt" INTO TABLE maildb;
ERROR: File 'users.txt' not found (Errcode: 2)
mysql> LOAD DATA LOCAL INFILE "d:\MYSQL\users.txt" INTO TABLE maildb;
ERROR: File 'd:MYSQLusers.txt' not found (Errcode: 2)
mysql> LOAD DATA LOCAL INFILE "d:/MYSQL/users.txt" INTO TABLE maildb;
Query OK, 2 rows affected (0.02 sec)
Records: 2 Deleted: 0 Skipped: 0 Warnings: 0
```

8. Inserting a single record/updating a field/encrypting passwords ...

(use 'NULL' where you don't have an entry for a field)

```
mysql> INSERT INTO maildb VALUES ('bob','bob','NULL');
Query OK, 1 row affected (0.01 sec)
```

oops should have encrypted the password ...

```
mysql> UPDATE maildb SET password = PASSWORD('bob') WHERE username = 'bob';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

so how do i add from text file and encrypt the passwords ...

I can't work out how to do this, so I had better encrypt all the other passwords ...

```
mysql> UPDATE maildb SET passwd = PASSWORD('tam') WHERE username = 'tam';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE maildb SET passwd = PASSWORD('test') WHERE username = 'test';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

now lets see what I have in my maildb database ...

```
mysql> SELECT * FROM maildb;

+----------+------------------+---------+
| username | password         | forward |
+----------+------------------+---------+
| tam      | 6752d6483e543e43 | NULL    |
| test     | 378b243e220ca493 | NULL    |
| bob      | 7d67547927a4589e | NULL    |
+----------+------------------+---------+
3 rows in set (0.01 sec)
```

oops i gave the wrong name to my table column 'password' ...

8. Changing a table column name ...

```
mysql> ALTER TABLE maildb CHANGE 'password' 'passwd' VARCHAR(20);
ERROR 1064: You have an error in your SQL syntax near "password' 'passwd' VARCH
AR(20)' at line 1
mysql> ALTER TABLE maildb CHANGE password passwd VARCHAR(20);
Query OK, 3 rows affected (0.19 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

now lets see what is in my maildb database again ...

```
mysql> SELECT * FROM maildb;

+----------+------------------+---------+
| username | passwd           | forward |
+----------+------------------+---------+
| tam      | 6752d6483e543e43 | NULL    |
| test     | 378b243e220ca493 | NULL    |
| bob      | 7d67547927a4589e | NULL    |
+----------+------------------+---------+
3 rows in set (0.01 sec)
```

Download and History for MySQLAuth

Generally you will find the latest download in with your distribution set. You can check what version of mysqlauth you have by running mysqlauth at a command prompt, e.g.,
c:\dmail\mysqlauth -version

If we put a download on the site for a version, it will listed in the history below as a link.

Here is the download for an example mysqlauth.ini file if you need it,
mysqlauth.ini

History:

1.0o 14 Aug 2000 (also in dmail 2.9a)
-fixed sql_del command using fixed username field.
mysqlauth10o_libc6

1.0n
-fixed death when no field_alias setting given.

1.0m (in dmail 2.8 versions)
mysqlauth10m_solarissparc.tar.Z

1.0g 28 June 2000
fixed responses so that a uid is always given, previously might respond, '+ok username path' which breaks E.A.P.

ODBCAuth External Authentication Module for DMail

An external authentication module that allows DMail to do user lookups on a Database with an ODBC driver.

Designed to work with, Microsoft SQL Server, Microsoft Access and ORACLE databases amongst others.

This is a Windows Only External Authentication Module.

Installing ODBCAuth

Once you have downloaded, ODBCAuth from the Utilities Download Page and unpacked it to your DMail directory (e.g. c:\dmail), you should find two files:

odbcauth.exe
odbcauth.ini

(If you cannot find these files contact, DMail Support)

Place these files into your dmail directory and update dmail.conf to point to this external module.

e.g. on Windows:
authent_method external
authent_process c:\dmail\odbcauth.exe

Then you must edit the odbcauth.ini file so that it points to the ODBC driver for your Database. More details on this step are provided on the rest of this page.

NB: You should locate the ini file in the same directory as the odbcauth binary or as, c:\winnt\system32\odbcauth.ini (on Windows platforms)

And then RESTART both DSMTP and DPOP.

NB: you must RESTART both DSMTP and DPOP when ever you change your odbcauth.ini file, as they spawn copies of ODBCAuth which only reads odbcauth.ini at startup.

And that is it. For details of other External Authentication Settings see the User Administration

section of the DMail Manual.

IMPORTANT Notes on using ODBCAuth

Notes for MS Access Database (Office 2000)
(Many will apply to all databases)

- have to save before columns are updated in odbc driver.
- you do not have to stop odbcauth when database changes, but sometimes you do have to stop editing the database - just saving it is NOT enough.
- you cannot save certain table changes in access when odbcauth is running, e.g. changing a column name, but you can add rows to a table or edit column values.
- spaces in column names get taken out so for column, Email Address, it should be identified in odbcauth.ini as, field_x EmailAddress
- space in table names do stay so,
  odbc_mail_user_table Mail Database
  is valid.
- (to be checked) any data type is ok in columns
- set odbc driver name with setting,
  odbc_driver_name odbcauth
- if odbcauth dies before connection to odbc driver then odbc_driver_name settting probably set incorrectly.

Creating/Using a ODBC Database

This section has still to be completed...

Basically you have to create your database or use an existing database.

Then setup an ODBC database driver to point to your database, in Control Panel, ODBC Data Sources.

Then setup ODBCAuth by editing its ini settings so that it talks to the odbc driver that you have set up.

Check Install

NB: ODBCAuth only supports the Basic External Authentication Protocol. So it only supports looking up a user (no password needed) and checking a user (requires username and password), and not adding, modifying or searching for users.

Once ODBCAuth is installed it is best to then check that ODBCAuth is working correctly. You will need to run ODBCAuth from the command line (in a dos box on Windows) and try the following. NB: We recommend that you run in debug mode at this stage. So you will get a lot of lines coming back

that are just information.

NOTE: C: - Client, S: - Server

c:\dmail\odbcauth -debug

C: lookup test
S: +OK test config 0

C: check test password
S: +OK test config 0

C: check test incorrect
S: -ERR test password wrong or not a valid user

The acutal response may vary. To display the complete list of commands that ODBCAuth supports enter the command:

help

Information about the protocol being used can be found in the DMail manual: (Basic External Authentication Protocol)

User Administration , External Authentication Protocol section

---

ODBCAuth.ini settings

Label	Example	Default	Explanation
debug (Optional)	true	false	No Longer settable in ini file, use 'log true' instead. Or if you really want extra info to 'stdout' then use the command line -debug option.
domain (Optional)	your.default.domain	none	This is default domain that is appended to any username that does not already have a domain setting attached.
field_username	user	username	The label of the username field in your table.
field_password	password	passwd	The label of the password field in your table.
field_forward (Optional)	fwd	none	The label of the forward field in your table. This is the fowarding information that DMail uses to determine if mail for the account looked up should be delivered to a different address.
field_quota (Optional)	quota	none	The label of the quota field in your table. This is the disk quota the user has. e.g. 100k, 10M

field_mailmask (Optional)	mailmask	none	The label of the mailmask field in your table. This is the IP mask that ODBCAuth checks against the from IP of the user. If specified, this forces the user to connect only to the specified IP in order to collect mail (POP).
field_maildrop (Optional)	maildrop	none	The label of the maildrop field in your table. This is the location where dmail will drop the user mail files. NB: if the field value is empty in the database (NULL) odbcauth will return the keywork 'config' indicating that the server knows where to locate the drop file.
log (Optional)	true	false	Makes odbcauth log verbose information to the log file.(error messages will always be logged)
log_path (Optional)	odbcauth.log	auth.log	This is the odbcauth log file.
odbc_login (Optional)	username	none	(Optional) This is the username that has access to the correct database and table that stores the usernames and passwords.
odbc_password (Optional)	password	none	(Optional) This is the password that is required and used in conjuction with odbc_login.
odbc_odbc_driver_name	user_data	odbcauth	This is the odbc driver name as set in ODBC Data Sources in Control Panel for the database where the mail usernames are stored.
odbc_mail_user_table	users_list	maildb	This is the table name that is within the mail user database that has the user details. NB: cannot have spaces in table name!
odbc_mail_uid (Optional)	99	0	This is the unique mail ID.NB: by default odbcauth returns 0 for this which indicates to the DMail Servers that the userid should not be checked.

IMAP Copyright Notice and Release Notes

```
 * Program: IMAP4rev1 server
 *
 * Author:  Mark Crispin
 *          Networks and Distributed Computing
 *          Computing & Communications
 *          University of Washington
 *          Administration Building, AG-44
 *          Seattle, WA  98195
 *          Internet: MRC@CAC.Washington.EDU
 *
 * Date:    5 November 1990
 * Last Edited:   8 September 1998
 *
 * Copyright 1998 by the University of Washington
 *
 *  Permission to use, copy, modify, and distribute this software and its
 * documentation for any purpose and without fee is hereby granted, provided
 * that the above copyright notice appears in all copies and that both the
 * above copyright notice and this permission notice appear in supporting
 * documentation, and that the name of the University of Washington not be
 * used in advertising or publicity pertaining to distribution of the software
 * without specific, written prior permission.  This software is made
 * available "as is", and
 * THE UNIVERSITY OF WASHINGTON DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
 * WITH REGARD TO THIS SOFTWARE, INCLUDING WITHOUT LIMITATION ALL IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND IN
 * NO EVENT SHALL THE UNIVERSITY OF WASHINGTON BE LIABLE FOR ANY SPECIAL,
 * INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
 * LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, TORT
 * (INCLUDING NEGLIGENCE) OR STRICT LIABILITY, ARISING OUT OF OR IN CONNECTION
 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.


imap-4.4 is a maintenace update.  It fixes the following bugs in imap-4.3,
which was distributed with Pine 4.03:
 1) international character searching did not handle MIME2 headers properly
 2) crashes if the first UID in a UID sequence range did not correspond to a
     message
 3) no messages selected if neither UID in a UID sequence range corresponded
     to a message
 4) memory leak in local news spool access each time the ACTIVE file was
     scanned for a valid news group.

This version supports IMAP4rev1 (RFC 2060).  There are major internal and
external changes from the IMAP2bis version (e.g. imap-3.6, distributed with
Pine 3.9x).  It is extremely unlikely that a program written for the IMAP2bis
```

verion will build with this version without modifications. Drivers written for the IMAP2bis version will definitely need to be rewritten.

Most notable operational changes from imap-3.x:

- SASL authentication is supported in the IMAP and POP3 servers, and in the IMAP, POP3, and SMTP client code. There is no support for NNTP SASL yet
- Kerberos V5 is supported through the GSSAPI on UNIX and NT. To enable Kerberos V5 on UNIX, add "gss" to the EXTRAAUTHENTICATORS list in the top-level Makefile and rebuild. You may also need to edit the GSSAPI directory paths in the ../src/osdep/unix/Makefile
- Kerberos V4 client-only contributed code is supplied, but is unsupported
- The mbox driver is now enabled by default. If the file "mbox" exists on the user's home directory and is in UNIX mailbox format, then when INBOX is opened this file will be selected as INBOX instead of the mail spool file. Messages will be automatically transferred from the mail spool file into the mbox file
  To disable this behavior, delete "mbox" from the EXTRADRIVERS list in the top-level Makefile and rebuild
- IMAP4rev1 protocol is now supported. The UNIX format support now maintains unique identifiers (UIDs) and keyword flags for each message, and keeps an invisible message at the start of the file which contains the UID base information and a list of assigned keywords. There is no way to disable this behavior, since it would disable IMAP4rev1 support. This message may show up if you access the mailbox as a file using older mail software (e.g. Pine 3.9x). It is invisible with IMAP or POP access, or with access as a file using Pine 4.0x.
- Support for additional mailbox formats
- No longer keeps entire mailbox in memory for UNIX format files
- Multilingual searching of the following charsets are supported:
  US-ASCII, UTF-8, ISO-8859-1, ISO-8859-2, ISO-8859-3, ISO-8859-4, ISO-8859-5, ISO-8859-6, ISO-8859-7, ISO-8859-8, ISO-8859-9, ISO-8859-10, ISO-8859-13, ISO-8859-15, ISO-2022-JP, ISO-2022-KR, ISO-2022-CN, ISO-2022-JP-1, ISO-2022-JP-2, GB2312 (alias CN-GB), CN-GB-12345, BIG5 (alias CN-BIG5), EUC-JP, EUC-KR, Shift_JIS, KOI8-R, KOI8-U (alias KOI8-RU), TIS-620, VISCII.
  All ISO-2022-?? charsets are treated identically, and support ASCII, JIS Roman, hankaku katakana, ISO-8859-[1 - 10], TIS, GB 2312, JIS X 0208, JIS X 0212, KSC 5601, and planes 1 and 2 of CNS 11643.
  EUC-JP includes support for JIS X 0212 and hankaku katakana
- Fast sorting including IMAP server-based sort
- Fast ordered-subject threading including IMAP server-based threading

Most notable programmer/external changes from imap-3.x:

- Additional ports
- New directory orientation, no separate non-ANSI sources
- New local file formats mbx and mx
- The bezerk driver has been retired, and replaced with the new unix driver which does not keep a snapshot in memory

. Many new added ports including NT (Win32 client, NT server)
. Many new data access functions
. Numerous interface changes; look at the .h files for details
. There are no known security problems in this version
. If you ignored the warnings and used the evil configuration file, its name and contents have changed incompatibly

---

Products Downloads Prices Support Company

Authentication Settings Tables:

For all examples below the the following apply:

- Main domain is netwin.co.nz with the IP address 1.2.3.4
- server name is mail.netwin.co.nz
- test user on the main domain is 'test'
- virtual domain if there is one is computers.com, with suffix of @computers.com **OR** IP address of 9.9.9.9
- test user on virtual domain is sales (sales@computers.com)
- prefix on virtual domain is 'cc_'
- **NB:** For domain specific settings in CWMail and NetAuth the settings go within a **'vhost'** section

List of Tables:

Tables comparing product settings

The following tables show you what settings equate to each other in each of the products' configuration file.

Table 1: Domain used in user database with authent_domain true:
(domain added onto the end of the username, as it appears in user database, e.g. nwauth.add)
NB: the '@' symbol is always implied within the user database

Config File + Domain:	Equivalent Settings:	Old Equivalent Settings:
dmail.conf (all domains)	authent_domain true	
dmail.conf (main domain)	host_domain netwin.co.nz (must be **first** host_domain setting)	host_domain netwin.co.nz dpop_host netwin.co.nz (pre dmail 2.7)
dmail.conf (suffix vdomain)	vdomain cc @computers.com computers.com c:\dmail\in\computers (3rd item in vdomain line = computers.com)	
dmail.conf (IP vdomain)	vdomain cc 9.9.9.9 computers.com c:\dmail\in\computers (3rd item in vdomain line = computers.com)	
cwmail.ini	N.A. - no setting for any domain	
netauth.ini (main domain)	domain netwin.co.nz	
netauth.ini (suffix vdomain)	domain computers.com	
netauth.ini (IP vdomain)	domain computers.com	

Table 2: Prefix used in user database (authent_domain false):
(prefix prepended onto the username, as it appears in user database, e.g. nwauth.add)

Config File + Domain:	Equivalent Settings:	Old Equivalent Settings:
dmail.conf (all domains)	authent_domain false	
dmail.conf (main domain)	no extra settings as no prefix on usernames for main domain	
dmail.conf (suffix vdomain)	vdomain cc @computers.com computers.com c:\dmail\in\computers vdomain_separator _ (1st item in vdomain line together with separator give 'cc_')	
dmail.conf (IP vdomain)	vdomain cc 9.9.9.9 computers.com c:\dmail\in\computers vdomain_separator _ (1st item in vdomain line together with separator give 'cc_')	
cwmail.ini	N.A. - no setting for any domain	

Authentication Settings Tables:

netauth.ini (main domain)	prefix NULL (tells netauth to put no prefix on main domain usernames, as no prefix on usernames for main domain)	
netauth.ini (suffix vdomain)	prefix cc_ (includes separator)	prefix cc prefix_separator _ (pre version 4, e.g. 3.0e, use 2 setttings)
netauth.ini (IP vdomain)	prefix cc_ (includes separator)	prefix cc prefix_separator _ (pre version 4, e.g. 3.0e, use 2 setttings)

Table 3: Suffix used for POP login (suffix based vdomain):
(How to set the suffix that the user should enter on the end of their username when they login to the POP server **directly** - with the settings below CWMail will put the suffix on for the user when they login with their normal username)

Config File + Domain:	**Equivalent Settings:**	**Old Equivalent Settings:**
dmail.conf (all domains)	no general setting	
dmail.conf (main domain)	no setting as main domain users do not use suffixes	
dmail.conf (suffix vdomain)	vdomain cc @computers.com computers.com c:\dmail\in\computers (2nd item in vdomain line is suffix = '@computers.com', NB: it includes separator character)	
dmail.conf (IP vdomain)	no setting as IP based virtual domain users do not use suffixes	
cwmail.ini	no setting as main domain users do not use suffixes	
cwmail.ini (suffix vdomain)	suffix @computers.com (includes separator chracter)	
cwmail.ini (IP vdomain)	no setting as IP based virtual domain users do not use suffixes	
netauth.ini (main domain)	no setting	
netauth.ini (suffix vdomain)	suffix @computers.com (includes separator)	suffix computers.com suffix_separator @ (pre version 4, e.g. 3.0e, use 2 setttings)
netauth.ini (IP vdomain)	no setting	

Examples:

1. **Just one domain, authent_domain true:**

   Login to DPOP with: test
   Login to CWMail with: test

   Name in user database: test@netwin.co.nz
   DSMTP authenticates: test@netwin.co.nz
   DPOP authenticates: test@netwin.co.nz

	dmail.conf	cwmail.ini	netauth.ini	pre 2.7 dmail.conf	pre version 4 netauth (3.0e)
Settings:	host_domain netwin.co.nz host_domain mail.netwin.co.nz authent_domain true	pophost netwin.co.nz smtphost netwin.co.nz	pophost netwin.co.nz domain netwin.co.nz	dpop_host netwin.co.nz host_domain netwin.co.nz host_domain mail.netwin.co.nz authent_domain true	domain netwin.co.nz domain_separator @
Defaults being used:					
Do NOT use these:	dpop_host netwin.co.nz or dpop_host mail.netwin.co.nz		prefix suffix domain_separator suffix_separator prefix_separator		

2. **Example 1 with 1 suffix based virtual domain added**
   (= main domain + 1 suffix vdomain + authent_domain true)

   2 users: test on netwin.co.nz and sales on computers.com

   Login to DPOP with: test and sales@computers.com from any ip address
   Login to CWMail at http://www.netwin.co.nz/scripts/cwmail.exe with: test
   Login to CWMail at http://www.computers.com/scripts/cwmail.exe with: sales

Authentication Settings Tables:

Name in user database: test@netwin.co.nz and sales@computers.com
DSMTP authenticates: test@netwin.co.nz and sales@computers.com
DPOP authenticates: test@netwin.co.nz and sales@computers.com

	dmail.conf	cwmail.ini	netauth.ini	pre 2.7 dmail.conf	pre version 4 netauth (3.0e)
Settings:	host_domain netwin.co.nz host_domain mail.netwin.co.nz authent_domain true vdomain cc @computers.com computers.com c:\dmail\in\computers	pophost netwin.co.nz smtphost netwin.co.nz vhost www.computers.com pophost computers.com smtphost computers.com suffix @computers.com vend	pophost netwin.co.nz suffix netwin.co.nz suffix_separator @ vhost www.computers.com pophost computers.com suffix @computers.com domain computers.com vend	dpop_host netwin.co.nz host_domain netwin.co.nz host_domain mail.netwin.co.nz vdomain cc @computers.com computers.com c:\dmail\in\computers	pophost netwin.co.nz suffix netwin.co.nz suffix_separator @ vhost www.computers.com pophost computers.com suffix @computers.com domain computers.com domain_separator @ vend
Defaults being used:					
Do NOT use these:	dpop_host netwin.co.nz dpop_host mail.netwin.co.nz host_domain computers.com		prefix prefix_separator domain_separator		

DList Email Commands

DList commands are commands the people on the lists (list subscribers) send to the list server. They are used to control which lists the person is joined to and to allow them to make use of other functions that lists offer.

How to use list Commands:

DList commands are written in the **body** of a normal email message. The email message is then sent to a special email address,

listname-**request**.

For example, for a list called 'trendsetters' and a list server running on the machine 'great.isp.com' the user would put commands within an email and send it to

trendsetters-request@great.isp.com .

(Actual messages to be distributed to other trendsetters on the trendsetter list would just be sent to trendsetters@great.isp.com)

NB: Many of DList's commands have synonym command names available. For example, instead of sending the command **join** the user could just send the command **subscribe** or even **sub**. Users may find it useful to use a synonym if they are using the command on other lists of which they are members.

NOTE to Moderators:

Many of the commands can take extra parameters (marked in square brackets), which are for use by list moderators. The command 'approve xxxx' should be given in the first line of a message, to signify to DList that the command that follows has come from a moderator.

The commands available are:

Command	Function
* parameters in square brackets are for the Moderator's use, other users leave them blank	(Most commands will result in the list server emailing the user back at their 'reply' address with the required information)

help	Returns the help information contained in these pages.
[approve password] join [address] (or add,subscribe,sub,signon,join)	Subscribes the user to the list. Moderators can subscribe other people by specifying 'approve xxxx' where 'xxxx' is the password.
[approve password] leave [address] (or remove,unsub,unsubscribe,signoff)	Removes the user from the list stated in 'list_name'. Moderators and users can unsubscribe other people (depending on access settings) Unsubscribe is also detected in messages sent to the list itself as this is a common mistake users will make.
who (or who,review,enumerate)	Sends a mail message to the user containing a list of **all** members of the list stated in the 'list_name' parameter.
lists	Returns a list of the lists handled by the list server, so the user can choose the one they'd like to join.
dir	Provides a list of receivable files available to users of the list 'list_name'

get <file-name> (or send)	Sends the files requested as attachments of an email message to the user. The User can put in as many 'file-name' parameters as needed in order to obtain all the files required. NB: Wildcards can be used in the filename.
approve xxxxx	For moderated lists, if a message contains on its first line approve 'xxxxx' then the item will be accepted as if it came from the moderator. (Assuming xxxxx is the correct password as defined in lists.dat)
digest true | false	If the user is set to 'digest true' then they will recieve all the day's messages at the end of the day in one mime message instead of as they arrive.
holiday true | false	When holiday is set true you will recieve no messages.
status	The sender will receive a message showing the status of the list.

Headers and Environment Variables

Under Construction :-)

1. Headers Added by DSMTP
2. Headers Added by DPOP
3. Headers Added by DList
4. Headers Added by DRespond
5. Environment Variables

1. Headers Added by DSMTP

   X-Rcpt-To: <destination_address_given_in_envelope>
   Received: from domain ([ipaddress]) by this_server ; Fri, 20 Aug 1999 10:49:19 +1200

   The following headers are only added if they don't exist already,
   Return-Path: senders_address_from_envelope(MAIL FROM:)
   Date: ddd, dd mmm yyy hh:mm:ss +/-timezone (timezone can be set in dmail.conf)
   Message-Id: <id@this_server>

   ---

2. Headers Added by DPOP

   X-DPOP: DPOP Version x.xx (can be hidden with x-dpop_header_hide true)
   Status: U, R, RO (only if it does not already exist)

   ---

3. Headers Added by DList

   X-Mailing-List: list_address
   X-listMember: address_of_poster [list_alias_used]

   ---

4. Headers Added by DRespond (auto-responder)

   X-Autoresponder: Drespond

   ---

5. Environment Variables

   DSMTP sets the following environment variables for any robot that it spawns.

   HOME- users nominal home directory,
   USER - username dsmtp had to lookup in user database
   MAILFROM - Raw SMTP envelope paramater for the MAIL FROM: line
   RCPTTO - Raw SMTP envelope paramater for the RCPT TO: line
   MSGSIZE - size of message(headers and body) in bytes written to drop file

   ---

RFC Compliance and Exceptions

This page lists some **but not all** of the RFCs that the DMail servers comply with.

In general all of the servers in DMail comply with the relevant sections of the relevant RFCs. We will try to note any exceptions in the lists below. Do feel free to contact DMail Support if you wish to query the behaviour of our servers or their RFC compliance.

DSMTP

- 821 SIMPLE MAIL TRANSFER PROTOCOL (SMTP)
- 822 STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES
  Exception 1: forward slashes '/' not supported in local usernames, but are supported for non-local users.
  Exception 2: escape characters not supported in local usernames, but are supported for non-local users.
  Exception 3: quoted usernames support added in version 2.7q

- 1651 (ESMTP) SMTP Service Extensions
- 1854 (ESMTP) SMTP Service Extension for Command Pipelining
- 1891 (ESMTP) SMTP Service Extension for Delivery Status Notifications
- 2487 (ESMTP) SMTP Service Extension for Secure SMTP over TLS
- 2505 Anti-Spam Recommendations for SMTP MTAs
- 2554 (ESMTP) SMTP Service Extension for Authentication

DPOP

- 1939 Post Office Protocol - Version 3

General

- 1321 The MD5 Message-Digest Algorithm
- 2195 IMAP/POP AUTHorize Extension for Simple Challenge/Response

---

Exceptions

1. Exception 1: We have decided that local usernames may not contain forward slashes '/', because of the ambiguity with drop file names. Connections giving a username containing a slash for a

local domain will get a "RCPT not OK" response. Version 2.7i and above passes on messages to non local domains even if they contain a forward slash.

2. Exception 2: Similarly we have decided that local usernames may not contain escape characters e.g. '\"'. Connections giving a username containing a slash for a local domain will get a "RCPT not OK" response as they will hit the exception above. Version 2.8n and above passes on messages to non local domains even if they ANY escaped character.
3. Exception 3: Versions 2.7q and up are compliant in respect to quoted usernames, e.g. "bob"@domainx.com will result in mail going to the local user bob if domainx.com is local and will be passed through unchanged if destined for a non-local domain.

Products Downloads Prices Support Company

General Index

This page has links to things that we consider you might not be able to find information on in any other way.

All dmail.conf settings should be looked for in one of the four pages,
Common Settings,DSMTP Specific Settings, DPOP Specific Settings or Dlist Settings

---

Contents:

Search: Enter Word(s) to find

Submit

Free Trial Updates

License Prices

Products Downloads

Support Company

Tellpop and Tellsmtp

Under Construction :-)

. . .

To load a new key - tellpop key xxx-xxx (no tellsmtp reload). . .

To register - tellpop register - creates regsiter.txt . . .

Tellpop commands

Tellsmtp commands

. . .

Controlling the DSMTP/DPOP Servers Following installation various options and settings can be adjusted to tailor DSMTP and DPOP to your specific requirements. You may also want to check on the current status of the servers to see how many connections are in use etc.

A command line utility is provided for such management tasks on each of the servers in addition to the settings in the config file, they are Tellpop and Tellsmtp. Of special note are the 'reload' commands that both utilities have which cause the server specific to that utility to reload the configuration file without having to restart, e.g. 'tellpop reload' makes DPOP reload the configuration file.

Alternatively both the config settings and the commands to the server can be managed with the DMAdmin utility which also controls all three servers.

DMAdmin:

DMAdmin is a graphical user interface for controlling DSMTP/DPOP/DList and their configuration settings. It runs on the Windows 95 or Windows NT platform. It is automatically installed and then started by DMSetup on these platforms. It can also be used to control a Unix version of DMail/DPOP remotely.

Tellsmtp Command Line Utility

A command line application. This runs on all systems and provides a quick and simple method of controlling and monitoring DSMTP.

Tellpop Command Line Utility

A command line application. This runs on all systems and provides a quick and simple method of controlling and monitoring DPOP.

NOTE 1: Initial installation and setup of DSMTP and DPOP is best handled by the DMSetup wizard. Tellpop, Tellsmtp and DMAdmin are for later fine tuning etc.

NOTE 2: Modifying a dmail.conf setting which is relevant to both DPOP and DSMTP, requires you to reload **both** servers individually with their reload configuration file commands, e.g. tellsmtp reload **and** tellpop reload. DMAdmin will do this automatically.

Products Downloads Prices Support Company

Wadduser - Example Web User Administration

Wadduser is Netwin's example Web based User Administration Program.

NOTE: This has been replaced and improved considerably by NetAuth which is also free with DMail. Please see, Setting Up a Web Based Email System with Auto Account Creation.

We have created it primarily as an add on to our example External Authentication Module, NWAuth to allow people to set up a Hotmail type of system, where users can add themselves and choose an email address for a web based email system.

Note: We have also now created a new product that is a replacement for wadduser which is just as customisable, has more functionality, but does not require as much setup. Its called NetAuth and it is free with DMail, you should see the following overview for more details,
Users Adding Themselves - like Hotmail

Wadduser is linked to NWAuth at build time and as such is an extended nwauth program which runs as a CGI.

On Windows platforms you will find nwauth.exe and wadduser.exe precompiled for immediate use in the distribution set that you download.

The Wadduser CGI is run from a normal web page, an example of which is provided in distribution sets on all platforms, wadduser.htm.

NB: You should modify wadduser.htm before use, to remove options that you do not want users to be able to do, e.g delete each other's accounts! :-) Click on this link to see an example wadduser page

For details on implementing Wadduser you should look at the following FAQs,
Technical setup details (How do I set up a 'Hotmail' type system?)
Adding fields to wadduser

Note: because wadduser is linked into nwauth if you ever upgrade nwauth then you should also upgrade wadduser to the same version.

Below is the source for Wadduser - see the file wadduser.c in your distribution set for the source applicable to your build of wadduser.

```
/*
See the instructions on
http://netwinsite.com/dmail/faq.htm

Basic CGI program to allow users to be added/removed/searched

If using this for users to register themselves, you should remove the web_del function.
```

External functions (in nwauth.c), compile nwauth.c with NOAUTHMAIN defined

```
int auth_exists(char *user);
int auth_set(char *user, char *pass, char *info);
int auth_del(char *user, int quiet);
void auth_search(char *user);
*/



#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#ifdef WIN32

#include <io.h>
#endif
#include <fcntl.h>

#ifndef TRUE
#define FALSE 0
#define TRUE (!FALSE)
#endif
#define BFSZ 300

char *lib_encode(char *kkkk);
char *value_encode(char *s);
char *mygetenv(char *v);
char *query_find(char *s);
int ispressed(char *s);
int query_get(void);
void getword(char *word, char *line, char stop) ;
char x2c(char *what) ;
char *email_name(char *email);
void unescape_url(char *url) ;
void plustospace(char *str) ;
void form_get(void);
int web_delete(void);
char *form_find(char *s);
char *ncpy(char *dst, char *src, int len);
int web_search(void);
void do_header(char *title);
int web_add(void);
void do_footer(void);
#include "nwauth.h"
```

```
char thisuser[200];
char method[200];
char *get_date(void);


int check_value(char *descr, char *symbol, char *dflt);
int main(int argc, char *argv[])
{
strcpy(thisuser,mygetenv("REMOTE_USER"));
strcpy(method,mygetenv("REQUEST_METHOD"));
form_get();
auth_init(); /* Init the nwauth functions */
if (ispressed("web_add")) web_add();
else if (ispressed("web_search")) web_search();
else if (ispressed("web_delete")) web_delete();
else web_add();
return 0;
}
void showfile(char *fname)
{
FILE *f;
char bf[BFSZ];
f = fopen(fname,"r");
if (f==NULL) return;
for (;!feof(f);) {
if (fgets(bf,BFSZ-1,f)==NULL) break;
printf("%s",bf);
}
fclose(f);
}
int web_add(void)
{
FILE *f;
char username[BFSZ],password[BFSZ],name[BFSZ];
char bf[BFSZ];
/* Check the user has filled in the required fields */
if (!check_value("Name","name","")) return 0;
if (!check_value("Username","username","")) return 0;
if (!check_value("Password","password","")) return 0;

f = fopen("adduser.log","a");
if (f==NULL) { printf("Could not write file\n"); return 0;}
fprintf(f,"%s|Add|",get_date());
fprintf(f,"%s|",mygetenv("REMOTE_ADDR"));
fprintf(f,"%s|",form_find("username"));
fprintf(f,"%s|",form_find("name"));
```

```
/* These are optional form elements to record */
fprintf(f,"%s|",form_find("phone"));
fprintf(f,"%s|",form_find("fax"));
fprintf(f,"%s|",form_find("comments"));
fprintf(f,"\n");
fclose(f);

ncpy(username,form_find("username"),BFSZ-1);
ncpy(password,form_find("password"),BFSZ-1);
ncpy(name,form_find("name"),BFSZ-1);

strlwr(username); /* Only allow lower case usernames */
do_header("Adding user");
printf("<pre>");
if (auth_exists(username)) {
printf("Sorry, a user by that name already exists\n");
} else {
sprintf(bf,"name=\"%s\"",name);
auth_set(username,password,bf);
showfile("added.htm");
}
printf("</pre>");
do_footer();
return 0;
}

int web_delete(void)
{
FILE *f;
char username[BFSZ];
/* Check the user has filled in the required fields */
if (!check_value("Username","username","")) return 0;

f = fopen("adduser.log","a");
if (f==NULL) { printf("Could not write file\n"); return 0;}
fprintf(f,"%s|Delete",get_date());
fprintf(f,"%s|",mygetenv("REMOTE_ADDR"));
fprintf(f,"%s|",form_find("username"));
fprintf(f,"\n");
fclose(f);

ncpy(username,form_find("username"),BFSZ-1);

strlwr(username); /* Only allow lower case usernames */
do_header("Deleting user");
printf("<pre>");
```

```
auth_del(username,FALSE);
printf("</pre>");
do_footer();
return 0;
}
int web_search(void)
{
char search[BFSZ];
/* Check the user has filled in the required fields */
if (!check_value("Search string","search","")) return 0;

ncpy(search,form_find("search"),BFSZ-1);
do_header("Search complete");
printf("<pre>");
auth_search(search);
printf("</pre>");
do_footer();
return 0;
}


/*--------- Worker routines -----------------------------------*/

void do_header(char *title)
{
printf("Content-type: text/html\n\n");
printf("<html>\n");
printf("<head>\n");
printf("<title>%s</title>\n",title);
printf("</head>\n");
printf("<body>\n<h3>%s</h3>\n",title);
}
void do_footer(void)
{
printf("</body>\n");
}

int check_value(char *descr, char *symbol, char *dflt)
{
if (strcmp(form_find(symbol),dflt)==0) {
fail:
do_header("Required field missing");
printf("Sorry, you must fill in all fields, you missed out <b>(%s)</b><p>\n",descr);
printf("Please click on 'back' and fill in the other fields, thanks.<p>\n");
do_footer();
return FALSE;
```

```
}
if (strcmp(symbol,"email")==0) if (strchr(form_find(symbol),'@')==NULL) goto fail;
return TRUE;
}

/* ------------------ library of stuff for web --------------------*/
char *get_date(void)
{
time_t t;
static char tstr[30];
t = time(NULL);
strcpy(tstr,ctime(&t));
tstr[13] = 0;
return tstr;
}

static char *query_name[1000];
static char *query_val[1000];
static int nquery;
/* Query string */
char *lib_encode(char *s)
{
char *ss=s;
for (;*s!=0;s++) {
if (isspace(*s)) *s = '+';
}
return ss;
}
char *value_encode(char *s)
{
 static char bf[2000];
 int i;
 char *out=bf;
 for (i=0; i<2000 && *s!=0;i++,s++) {
  if (*s=='<') { strcpy(out,"<"); out += strlen(out); }
  else if (*s=='>') { strcpy(out,">"); out += strlen(out); }
  else if (*s=='&') { strcpy(out,"&"); out += strlen(out); }
  else *out++ = *s;
 }
 *out++ = 0;
 return bf;
}



char *mygetenv(char *v)
{
```

```
char *s = getenv(v);
if (s==NULL) return "";
return s;
}
char *query_find(char *s)
{
int i;
for (i=0; i<nquery; i++) if (strcmp(query_name[i],s)==0) return query_val[i];
return "";
}
int query_get(void)
{
char *q;
char *cl;
char bf[10000],val[2000],name[BFSZ];
int x;

q = mygetenv("QUERY_STRING");
if (q==NULL) return FALSE;
strcpy(bf,q); cl = bf;
for(x=0;cl[0] != '\0';x++) {
getword(val,cl,'&');
plustospace(val);
unescape_url(val);
getword(name,val,'=');
query_name[x] = strdup(name);
query_val[x] = strdup(val);
/*printf("[%d] {%s} = {%s} \n",x,query_name[x],query_val[x]);*/
}
nquery = x;
return TRUE;
}
void getword(char *word, char *line, char stop)
{
int x = 0,y;

for(x=0;((line[x]) && (line[x] != stop));x++)
word[x] = line[x];

word[x] = '\0';
if(line[x]) ++x;
y=0;

while(line[y++] = line[x++]);
}
char x2c(char *what)
```

```
{
register char digit;

digit = (what[0] >= 'A' ? ((what[0] & 0xdf) - 'A')+10 : (what[0] - '0'));
digit *= 16;
digit += (what[1] >= 'A' ? ((what[1] & 0xdf) - 'A')+10 : (what[1] - '0'));
return(digit);
}

void unescape_url(char *url)
{
register int x,y;

for(x=0,y=0;url[y];++x,++y) {
if((url[x] = url[y]) == '%') {
url[x] = x2c(&url[y+1]);
y+=2;
}
}
url[x] = '\0';
}

void plustospace(char *str)
{
register int x;

for(x=0;str[x];x++) if(str[x] == '+') str[x] = ' ';
}



static char *form_name[1000];
static char *form_val[1000];
static int nform;
char *makeword(char *line, char stop) ;
char *fmakeword(FILE *f, char stop, int *cl);
void convert_crlf(char *s); /* turn crlf into lf's */
void form_get(void)
{
int x,cl;
char *s = mygetenv("CONTENT_LENGTH");
if (s==NULL) return;
cl = atoi(s);
for(x=0;cl && (!feof(stdin));x++) {
form_val[x]= fmakeword(stdin,'&',&cl);
plustospace(form_val[x]);
unescape_url(form_val[x]);
```

```
form_name[x] = makeword(form_val[x],'=');
convert_crlf(form_val[x]);
}
nform = x;
}
void convert_crlf(char *s)
{
char *out=s;
for (;*s!=0;s++) {
if (*s == '\015') s++;
*out++ = *s;
}
*out++ = 0;
}
char *makeword(char *line, char stop)
{
int x = 0,y;
char *word = (char *) malloc(sizeof(char) * (strlen(line) + 1));

for(x=0;((line[x]) && (line[x] != stop));x++)
word[x] = line[x];

word[x] = '\0';
if(line[x]) ++x;
y=0;

while(line[y++] = line[x++]);
return word;
}

char *fmakeword(FILE *f, char stop, int *cl)
{
int wsize;
char *word;
int ll;

wsize = 8024;
ll=0;
word = (char *) malloc(sizeof(char) * (wsize + 1));

while(1) {
word[ll] = (char)fgetc(f);
if(ll==wsize) {
word[ll+1] = '\0';
wsize+=8024;
word = (char *)realloc(word,sizeof(char)*(wsize+1));
```

```
}
--(*cl);
if((word[ll] == stop) || (feof(f)) || (!(*cl))) {
if(word[ll] != stop) ll++;
word[ll] = '\0';
return word;
}
++ll;
}
}



char *form_find(char *s)
{
int i;
for (i=0; i<nform; i++) if (strcmp(form_name[i],s)==0) return form_val[i];
return "";
}
char *form_find_i(char *s, int i)
{
char bf[BFSZ];
sprintf(bf,"%s%d",s,i); s = bf;
for (i=0; i<nform; i++) if (strcmp(form_name[i],s)==0) return form_val[i];
return "";
}

char *email_name(char *email)
{
static char bf[BFSZ];
char *s2;
char *s;
s = strchr(email,'(');
if (s!=NULL) {
strcpy(bf,s+1);
s = strchr(bf,')');
if (s!=NULL) *s = 0;
}
ncpy(bf,email,BFSZ-1);
s = strchr(bf,'@');
s2 = strchr(bf,' ');
if (s==NULL) s = s2;
if (s2!=NULL) if (s2<s) s = s2;
if (s!=NULL) *s = 0;
return bf;
}
```

```
char *towww(char *s)
{
static char bf[BFSZ],*out;
for (out=bf;*s!=0;s++) {
if (isspace(*s)) *s = '+';
if (*s=='+') {sprintf(out,"%%%x",*s); out += strlen(out);}
else if (*s=='&') {sprintf(out,"%%%x",*s); out += strlen(out);}
else *out++ = *s;
}
*out++ = 0;
return bf;
}
char *ncpy(char *dst, char *src, int len)
{
char *xdst = dst;
int xlen = len;
for (;(*src !=0) && (len>0); ) {*dst++ = *src++; len--;}
*dst++ = 0;
xdst[xlen] = 0;
return xdst;
}
int ispressed(char *s)
{
char bf[BFSZ];
if (strlen(form_find(s))>0) {
return TRUE;
}
sprintf(bf,"%s.x",s);
if (strlen(form_find(bf))>0) {
return TRUE;
}

return FALSE;
}
```

User Administration Example Page

- This is a non-working example, but do use the 'view source' option on your browser to see for yourself that the form on this page is plain HTML :-)

Enter users full name

Enter desired username

Enter password

Add User

Delete User

Search

Notes on DMSetup Questions

Warning this page is still under construction !!!

Given that you are using the TCPIP protocol across your network as well as any other protocols I suggest that you basically accept most of the defaults(simply press return - the default is in the square brackets), e.g. for the directories to install into etc. If you are not using TCPIP then get back to me and I can tell you how to set it up.

Check that the you answer the question, What is the TCPIP name of this computer [your machine] ? with the machine name or IP address for the computer running DMail.

Set the email address for system administrator?
question to a valid email address where you want to receive all of the messages to do with the email server administration.

Another local domain for email [ ] ?
answer this with any other TCPIP IP addresses or IP names for your machine (note this does not include any virtual domains that you wish to host). If you specified an IP address for the machine name of your computer then you might want to enter the matching IP name here.

Path for mail drop files [C:\dmail\in] ?
accept default

DPOP creates a user.bin directory for each user for unread msgs etc.
Do you want to store these in the same directories as drop files [yes] ?
accept default

Should DMAIL use a gateway when sending email to external users [no] ?
accept default

Do you want to place limits on user access to the DPOP mail server [no] ?
accept default

Password for manager control of DSMTP/DPOP/DLIST [1729327945] ?
I reccomend setting this to something easy like "password" rather than the default number for your situation. This password is needed only when you wish to administer this DMail Server remotely with the DMAdmin GUI utility from another machine. It is not needed to administer the server using DMAdmin when on the same machine.

Limit from-address for manager control to [*] ?
accept default

Do you want to change any of your answers [no] ?
at this point you can go back and change any of your answers by pressing 'y' - if you do go back your previous selections will now appear as the defaults.

DMSetup will now copy files and generally do setup things for you. If you are not running on Windows NT then you will probably see error messages about failing to start DMail as a "service" - you can

ignore these.

Shall I start DSMTP/DPOP/DLIST now [yes] ?
select yes

I observed on my test machine the following perfectly normal messages:

Stop service failed 998
Successfully deleted service {dwatch}
Installing service {dwatch} to start image {C:\dmail\dwatch\dwatch.exe}
Services dwatch successfully added to local database
Service started ok
START service failed 1056

You should then see DMail Manager (also known as DMAdmin) start after a few seconds. If all of the servers are working there should appear a green bar above each of the tabs.

To add users to your email system you will probably have to use external authentication. There is a sample program which you can modify if you wish, nwauth.exe included in the distribution set.

You will need to open up the dmail.conf text file (which you will find in \winnt\system32 directory on NT and in /etc on Unix based systems) with a text editor, e.g. notepad or vi and type in the following two lines,

authent_method external
authent_process c:\dmail\nwauth.exe

Then you should use DMAdmin to stop and restart both DPOP and DSMTP.

Installing and Running DMail on Windows 95 or 98

Under Construction :-)

DMail can be run on the Windows 95 and Windows 98 platforms. As these platforms do not have 'Services' as on Windows NT, a few things have to be done a little differently, particularly by the installation utility, DMSetup.

Installing DMail on Win 95/98

You should not have any problems installing DMail on windows 95/98. DMSetup will detect that you are on Win 95/98 and do the installation (or upgrade) appropriately. From time to time you may see messages indicating that a 'service' could not be started, simply ignore these as they only apply to NT services. We have tried to eliminate them as much as possible.

DMail must run using an external authentication system, as there is no operating system user database/password list which it can use. Netwin provides NWAuth, an external authentication module, free with DMail. DMSetup will install this automatically for you.

Running DMail on Win 95/98

As pointed out in the installation section above, you will have to run an external authentication module. DMSetup will have installed NWAuth, but you can change to something else if you wish, e.g. LDAP authentication. See the User Administration section for more details.

The other major thing to remember is that you should not use CTRL-C to stop any of the servers or DWatch. If you do this then any programs that they have spawned, e.g. NWAuth cannot be stopped without doing a restart.

So to stop the servers, you should create a file called, server_name.exit in the dwatch directory. Where server_name is the program name, i.e. dpop,dsmtp,dlist,dwatch, and the dwatch directory is set in your dmail.conf (\windows\system\dmail.conf) file by the dwatch_path setting. The default dwatch directory is
\dmail\dwatch.

E.g. to stop dwatch copy an existing text file into the dwatch directory with the file name, dwatch.exit

and in a few seconds you should see it find the file and shut itself down. For example, you could do the following,
cd \dmail\dwatch
copy dwatch.pid dwatch.exit

If you have any problems or queries please contact dmail support at,

---
