Mercury/32 Help Index

Mercury is a Mail Transport System - a program that deals with the low-level mechanical process of moving electronic mail from place to place. While primarily intended for use with Internet Electronic Mail conforming to the RFC821 and 822 standards, it is nonetheless a general-purpose system capable of a wide range of unattended functions.

Key topics covered in this help file:

An overview of the Mercury System
Your license to use the Mercury/32 software

Configuring the Mercury core module
Configuring mailing lists
Using mailing lists
Configuring the Mercury Mail Server (Maiser)
Template files
Aliases, synonyms and autoresponders
Setting up automatic replies
Automated mail filtering
Sending messages from within Mercury
Locking the console

MercuryS, SMTP Server module
MercuryC, SMTP Client module
MercuryE, Full SMTP Delivery Client
MercuryP, POP Server module
MercuryD, POP3 Client module
MercuryX, Connection Scheduler module
MercuryH, PH Directory Server

Support for Novell NetWare Local Area Networks
Daemons (third-party extensions)

Obtaining Mercury Manual sets

Mercury - an overview

Mercury is divided into two principal sections:

The Mercury Core Module This part of Mercury is contained in MERCURY.EXE itself, and is responsible for distinguishing between local mail and foreign mail, for ensuring that messages are either delivered correctly if local, or routed to the proper protocol module if foreign, and for providing the core functionality of the system, such as the mail server, distribution list management, automatic replies and autoforwarding.

Mercury Protocol Modules Protocol modules are DLLs loaded by Mercury that provide services to the core module. Protocol modules are generally tied to a specific delivery protocol and are usually used to ferry mail in and out of the Mercury system. Nine protocol modules are supplied with the Mercury System:

MERCURYS.DLL	An SMTP Server module for handling incoming SMTP mail
MERCURYC.DLL	An SMTP Client module, for sending outgoing SMTP mail
MERCURYE.DLL	A full SMTP delivery client module for outgoing SMTP mail.
MERCURYP.DLL	A POP3 server, to allow POP3 clients to retrieve mail from the Mercury system.
MERCURYF.DLL	A simple Finger server
MERCURYD.DLL	A POP3 client capable of retrieving mail automatically on behalf of users on the local system.
MERCURYX.DLL	A scheduling module, allowing co-ordinated startup and shutdown of the various Mercury modules. Invaluable for use in dial-up environments.
MERCURYH.DLL	A PH Query Server for Directory Services.
MERCURYW.DLL	A server that allows users to change their passwords.

MercuryE and MercuryC both provide the same functionality but in different ways: MercuryE delivers directly to the recipient's mail server, while MercuryC uses a technique called Relaying, by asking another single computer to send all mail on its behalf. MercuryC is ideally suited for use behind firewalls and on dialup links where the shortest connection time is required.

Core functionality

Mercury incorporates the following key features:

Mail filtering You can create sets of filtering rules that can perform arbitrarily complex processing on your incoming and outgoing mail. Rules can be global or general - global rules are applied to all mail, while general rules are tied to a single address. Mail filtering is enormously powerful - for more information, click here.

Mail server Mercury's mail server allows users to perform many actions automatically by sending mail to the reserved address ***Maiser*** on your system. The functions supported by the mail server include user lookup (the LOOKUP and VERIFY commands), subscription and unsubscription from public mailing lists (SUBSCRIBE, UNSUBSCRIBE, ADD, REMOVE), file transfer (the SEND command) and deferred mail options. For more information on configuring and using the mail server, click here.

Mailing list support Mercury has extremely powerful mailing list support, with a rich range of options. For more information on mailing lists, click here.

Native network integration Mercury has a comprehensive plugin interface allowing it to take advantage of underlying network architectures, such as Novell NetWare. In native Network mode, Mercury can support concepts such as Network groups, automatic mail directory location and configuration, and automatic use of the Network's user database.

Comprehensive alias and autoresponder support Mercury has powerful aliasing features, supporting a practically unlimited number of aliases that can be edited and modified while the program is running. Through aliases, you can also create powerful *autoresponders* - specially-formatted messages returned automatically in response to mail sent to specific addresses on your system. For more information on using aliases and autoresponders, click here.

Exceptional performance Mercury has been developed to be as fast as possible without introducing noticeable load on the Windows system on which it runs.

Maximum flexibility with its plug-in protocol modules, plug-in native network support layer and scheduling module, Mercury can be adapted for use in practically any environment.

Locking the Mercury Console

In some environments, you may not want Mercury's configuration dialogs to be accessed by unauthorised users while the program is running. You can offer yourself some protection from this kind of tampering by selecting *Lock console* from the *File* menu. Mercury will prompt you to enter a password twice, and when you have done so and pressed *Enter*, will no longer allow any access to its menus or permit itself to be terminated without the password. Once the correct password has been entered, the console is unlocked and operation continues normally. While the console is locked, the program may be minimized and restored normally, allowing the console information to be viewed by anyone.

Note: Mercury cannot prevent itself from being terminated by the Windows Task Manager (accessed by pressing Ctrl+Alt+Del). In environments where this is an issue, you should consider using the Microsoft Policy Management software to disable the Task Manager's "Terminate Application" option.

Configuration

Mercury is a large, rich system and offers a wide range of configuration options, accessible via the *Configuration* menu.

Mercury Core Module
Mailing lists
Template files
The Mail Server
Mail filtering rules

You can also configure individual Protocol Modules from the Configuration menu - each Protocol Module will have its own online help you can consult after you have opened its configuration dialog.

The Mercury Core Module Configuration

To configure the basic operation of the Mercury core processing engine, choose *Mercury core module* from the *Configuration* menu.

A tabbed dialog will appear offering several pages you can select. The items on the front page of the dialog, General, are described below. The other pages are:

Local domains	Tells Mercury how to identify local addresses
Groups	Tells Mercury about groups on your Network
Files	Tells Mercury where to find or put its files
Reporting	Controls statistics and system messages

Options on the *General* tab

Internet name for this system Enter here the Internet name for the machine on which Mercury is running. Mercury will use this information when forming certain addresses, such as the postmaster address. The name you enter here should be a fully-qualified domain name; if you are intending to use Mercury to provide mail services outside your immediate organization, the name you provide will need to be accessible in your Domain Name Server (DNS) system.

Mail queue directory, SMTP queue directory These entries control where Mercury should look for and place mail that is to be processed. The mail queue is where mail clients such as Pegasus Mail put messages for Mercury to process; Mercury also places jobs here on occasions, usually when generating autoreplies and mailing list mail. The SMTP queue is the location where the Mercury Core Module should place messages intended to be sent to the outside world by the MercuryC or MercuryE Client module. The mail queue and SMTP queues can be, and normally are the same.

Local mailbox directory path (This entry is ignored if you are using a network support module). Tells Mercury how to locate your users' mailbox directories. The string is a standard pathname containing one of two special placeholders - either *~8* or *~N*. When Mercury uses the string to find the mailbox for a user, it replaces *~8* with the first eight characters of the user's name, or replaces *~N* with the user's whole username. If you are using Pegasus Mail v3.01d or earlier, or any 16-bit Pegasus Mail client as your mail client in conjunction with Mercury, you should not use the *~N* substitution - you should only use the *~8* version. If this 8-character restriction creates problems with usernames for you, you could consider defining synonyms for the names that are longer than 8 characters, or upgrading to a later version of Pegasus Mail.

NOTE: It is currently a restriction of Mercury that the ~8 or ~N placeholder must appear at the end of the path - so, *C:\PMAIL\~8* is legal, but *C:\PMAIL\~8\MAILBOX* is not.

Time zone Enter here the timezone for your site, expressed as a plus or minus difference from GMT. So, if you are in Los Angeles and are currently at GMT - 9 hours, you would enter -0900 in this field. Mercury will accept the so-called "vernacular" time zone format, such as PST and CST, but the use of these formats is no longer recommended on the Internet and we strongly advise you to avoid them, since their use makes it impossible for most mail programs to sort properly by date.

Poll mail queue every x seconds This setting controls how often the core module should check to see if there is mail waiting to be processed in the queue. We recommend that you do not set it below ten seconds for performance reasons.

Username of postmaster Every system capable of receiving Internet mail must have a user called *postmaster*, to whom problem and status reports are sent. The postmaster account is usually an alias to a real user on your system, and this is the expectation within Mercury. Enter in this field the username of the user on the machine where Mercury is running who is to act as your postmaster. While it is

permissible to have a non-local address as your postmaster address, we strongly recommend you do not do this, since it can create real problems and mail loops when the remote machine is unreachable. This setting is mandatory - Mercury cannot run properly without it.

For delivery failures return x lines of the message When Mercury cannot deliver a message to a local user for whatever reason, it will invoke a template file you provide for delivery failures. One of the optional replacements that can be used in the delivery failure template file is a special substitution that sends a certain number of lines from the failed message. This configuration option controls how many lines of the message are returned when the special partial return substitution is encountered.

Broadcast notifications for normal mail Mercury has special Network awareness modules that allow it to take advantage of certain specific features of some local area networks. One of the features that some networks (such as Novell NetWare) support is the transmission of a single-line broadcast message that appears on the target user's screen. If this control is checked and you are running Mercury on a network that supports broadcast messages, Mercury will send a short message to users when new mail arrives for them.

Broadcast notifications for receipts (See the preceding section for more detail) This control determines whether Mercury should send broadcast messages advising the arrival of mail messages that confirm reading or delivery.

Send copies of all errors to the postmaster If this control is checked, Mercury will send a copy of all error reports it generates to the local postmaster as well as to the original sender of the message. This allows the postmaster the option of correcting addressing errors and other simple problems.

Change file ownership to recipient As with broadcast notifications, some Network systems support the idea of file ownership, usually to calculate disk space usage. If your network supports this idea and this control is checked, then Mercury will attempt to change the ownership of all the messages it delivers so that the actual recipient owns the file.

Suppress validation of From field when processing mail Mercury usually attempts to validate that the "From" field of all mail it delivers is legal. This can sometimes cause problems if you receive mail from sites that use broken or faulty mail programs; if this is the case, you can suppress the validity check Mercury performs by checking this control.

Hard to quit (exit only on Ctrl+File|Exit) When this option is checked, Mercury will ignore all attempts to quit from it, and will minimize itself to the system tray instead. In order to quit from the program, choose "Exit" from the "File" menu while holding down the Ctrl key. This option is useful when Mercury is run on a server to prevent people from accidentally closing it down.

Configuring local domains

See also: Core module configuration

Domains recognized as local by this server This is probably the single most critical area of configuration in the Mercury system -- if you get this section wrong, you will inevitably get mail loops and other problems. In this section, you must tell Mercury all the Internet names it should regard as "local" -- that is, for which it should attempt direct delivery on the local system rather than forwarding the mail to another machine for processing.

The *host/server* section of each definition is intended to allow Mercury to deliver mail to multiple file servers in supported network environments: if you are running Mercury on a single system or serving Pegasus Mail in either networked or multi-user standalone mode, the host/server entry is ignored. In the NetWare Bindery mode environment, this part is used to tell Mercury that a particular domain represents addresses on a specific file server or tree. In the NetWare NDS mode environment, this part is used to tell Mercury that a particular domain represents addresses within a specific segment of your NDS tree (see below for more detail).

When entering domains into this section, you should usually provide three entries per local Internet domain - a fully-qualified version, a simple version, and a special entry called a domain literal version, which is the IP number of your system enclosed in square brackets. For example, if your system's Internet name was calliope.pmail.gen.nz (192.156.225.76), you might create these domains definitions:

calliope	calliope
calliope	calliope.pmail.gen.nz
calliope	[192.156.225.76]

Domain mailboxes Mercury supports the idea of a domain mailbox, or a mailbox that accepts mail addressed to any user at a given domain. To create a domain mailbox, first create the user account that is to receive all mail addressed to the domain, then place an entry in the *Domains recognized as local by this server section* in the following format:

DM=username domain address

username can be any valid reference to a single local user on your system. So, to create a domain mailbox where user *mailserver* receives all mail addressed to any user in the domain *fish.net*, you would create this entry:

DM=mailserver fish.net

With this entry in place, mail sent to *[any address]@fish.net* will be delivered into user *mailserver*'s mailbox.

NDS Mode In NetWare NDS mode, the domains section can be used to tie a domain to a specific portion of your tree. So, if you have all mail sent to the domain `myorg.com` to a context in your NDS tree called `sales.us.myorg`, you would use this entry:

```
[Domains]
sales.us.myorg : myorg.com
```

When specifying an NDS domain, you can apply the definition to an entire portion of a tree (including all sub-levels within the NDS tree) by prefixing the context name with the special character */* - so, in the example above, if you simply wanted to equate your entire NDS tree with the domain `myorg.com`, you would use this entry:

```
[Domains]
/[root] : myorg.com
```

Domain Name Service The process by which systems connected via the Internet find out how to contact each other. Each organization on the Internet advertises the names of its machines via a primary Domain Name Server (usually a unix system).

Configuring Mailing Lists

Mercury has strong support for mailing lists -- groups of addresses that can be associated with a single address on your system. When a mail message is sent to the address associated with the list, Mercury will send it on to everyone who has subscribed to the list. Mercury's mailing lists are created and managed using the *Mailing lists* option on the *Configuration* menu.

Using mailing lists - commands and operation

Mailing lists have three key elements: Membership, Moderators and Settings.

Membership: The membership of a mailing list is the group of people who are subscribed to it at any given time. Mercury's mail server allows people to subscribe and unsubscribe automatically by sending it messages containing subscription commands. List members also have a certain amount of control over the way they receive mail -- they can choose to enable and disable receipt of mail from the list, and if you have enabled digest support for a list, they can choose whether or not they want to receive their mail in digest format.

Moderators: A mailing list can have one or more moderators, who are effectively managers for the list. Moderators have full control over the membership and settings of a list, and you can also configure a list so that only moderators may actually send mail to its membership: when you configure a list this way, then the list is said to be *moderated* - that is, only specific people can send mail to it. The intention of a moderated mailing list is that mail must be submitted to the moderator, who will then decided if it should be distributed. Note that a list can have moderators without being a moderated list - that is, a list can have supervisors, but can still distribute mail sent from the general public. A list need not have any moderators if you wish, and it is permissible for a moderator not to be a member of the list.

Settings: Mercury offers a wide range of settings that can be applied to a mailing list, which control the way it behaves when it receives mail for distribution, and the way it responds to control requests, such as subscription messages. The following is a summary of the mailing lists settings supported by Mercury and how to enable or disable them.

List title Every list must have a title -- a descriptive name that Mercury will use to form the "from" field of messages sent to the list. Try to keep the title short and descriptive and avoid international or accented characters. On rare occasions, you may wish to include address details as part of the title (Mercury usually adds the proper address to the list title automatically): in this case, you should ensure that the address you enter conforms to RFC822 addressing rules and includes a fully-qualified domain address appropriate for the list, then check the control labelled *Is a full address* next to the list title. *NOTE:* This feature is extremely specialised and is not normally required; because it can cause problems with mail delivery, we recommend that you only use it if you are very sure of what you are doing.

Public vs Private lists A public list will accept subscription requests from anyone who sends the proper command to the Mercury mail server. A private list will only accept subscription requests from people who are moderators of the list. To allow public subscription to a mailing list, click the *Allow public subscription* checkbox in the mailing list definition dialog.

Welcome files, farewell files You can create simple text files that are automatically sent when someone subscribes or unsubscribes from a mailing list. These files usually contain instructions for unsubscribing and resubscribing to the list, but can contain anything you feel is appropriate. Welcome and farewell files are entered in the respective fields in the mailing list definition dialog.

List signatures A list signature is a small text file that is automatically appended to the end of every message distributed to the list membership. In digest mode, the list signature is appended once as a separate message at the end of the digest. The first line of the list signature must contain the text to be placed in the "Subject" field of the digest part; the remainder of the signature can be whatever text you

wish to include. The "subject" line is ignored for non-digest subscribers. List signatures are usually used to include information on unsubscribing from the list, or on contacting the list moderator. They are optional - if you do not want to define a list signature, leave this field blank.

Error and reply routing If you enter an address in the text field labelled *Errors go to*, then Mercury will format the list mail in such a way that properly-written Internet mail transports will send messages concerning problems with delivery to that address. Unfortunately, not all Internet mail transports handle this situation correctly, so in some cases delivery problem notifications will be sent either to the list or to the original sender even if this field is properly sent. We strongly recommend that you supply an address in this field for all your lists. If you want replies to list messages to be sent to the list instead of to the original sender, click the checkbox labelled *Route replies to the list*.

Archive file Mercury can save copies of every message sent to a list in an archive file. If you want it to do this, enter an archive filename here. The filename must be a legal DOS filename and can include a path if you want to create it in a specific directory. You can use the following special characters in the filename:

~Y The year, expressed as two digits
~M The month, expressed as two digits
~D The day, expressed as two digits
~W The week of the year, expressed as two digits.

Using these substitution characters allows you to create sequences of archive files matching specific periods of activity.

Controlling mail submission You can specify that only list moderators may send mail to the mailing list by clicking the *Moderator submission only* checkbox in the mailing list definition dialog. You can also set the list to accept mail only from current list members by clicking the *Member submission only* checkbox. If neither of these controls is checked, then anyone may send mail to the mailing list. Finally, you can control the maximum allowable size of submissions to the list using the *Size limit* control: if a value greater than 0 is entered in this field, then any attempt to send a message larger than that number of bytes to the list will fail with an error.

Headers and URLs If the *Use helper URL headers* option is turned on, Mercury will add specially-formatted headers to messages distributed to the list which will permit compliant mail programs like Pegasus Mail to perform automatic subscription management for the user. If you have a web page that describes the operation of the mailing list, enter its URL in the *Help URL* field. Using Helper headers and URLs can result in a considerable improvement in the usability and friendliness of your lists.

Concealing a list The Mercury Mail Server, MAISER, has a command (LIST) that returns all the lists serviced by the running copy of Mercury. If you do not want a list to appear in responses to this command, check the control labelled *Conceal from Maiser LIST command* and it will not be shown.

Anonymous mail support It is occasionally desirable to set up mailing lists that provide anonymity for people who send mail to them -- examples of this include suggestion boxes, and lists covering sensitive or dangerous subjects. Mercury lists support three levels of anonymity - none, where no attempt is made to hide the sender's identity; logged, where no indication of the sender's identity appears in mail sent out to the membership, but the sender's address is recorded in the Mercury log file; and total, where the sender's identity is neither shown in mail sent to the list nor in the Mercury log file. ***WARNING:*** In many states and countries, there may be legal issues associated with hosting an anonymous list, particularly if it involves discussion of activities that are illegal or subversive. Before agreeing to host an anonymous mailing list, we strongly recommend that you consult your legal obligations.

Digest support Mercury has comprehensive support for mail digests. To enable digests support for a mailing list, enter a simple filename in the Digest filename field of the Mailing list definition dialog. The filename you enter may not have a path component - it is always stored in the Mercury scratch directory. The filename may not contain substitution characters the way an archive filename does. If no filename is entered in a list definition, then digest support will not be available for that list. If you want new subscribers to the mailing list to be subscribed automatically in digest mode, click the checkbox labelled

New subscribers default to digest mode. You cannot prevent a subscriber from changing in or out of digest mode if digest support is enabled for a list. The *Max size* and *Max waiting period* fields control the trigger conditions that determine when a digest is sent out to digest subscribers. If the *Max size* field is non-zero, then the digest will be distributed as soon as the digest file exceeds the number of bytes you enter. If the *Max waiting period* field is non-zero, then the digest will be sent after that number of hours has elapsed.

Passwords A password or passwords can be associated with a mailing list. When this is done, commands that can only be issued by moderators will need the password before they can be processed. The password is supplied by issuing a PASSWORD command in the message to the mail server at some point in the message prior to the command that needs it. So, if you have set the password fubar on the list called vobis on your server and a moderator wants to add a user to that list, he or she will need to send something like this:

```
password fubar
add vobis user@host.domain
```

You can provide either a single password for a mailing list, or a file of passwords. If you provide a file of passwords, then any password in the file can be used to gain access to the list. This latter approach allows you to give each moderator his own password, and revoke it without affecting other moderators in the event that he or she ceases being a moderator.

Digests: a digest is a single mail message that contains a group of other mail messages. For busy mailing lists, it is often convenient to allow mail sent to the list to accumulate in a digest and be sent out periodically, instead of sending the messages out immediately.

Mercury supports a special digest format called the MIME digest format. MIME is an Internet standard for the composition and presentation of messages and is widely supported. If your subscribers use a mail program that supports MIME digests, such as Pegasus Mail, then digests generated by Mercury will look like a kind of mail folder to them and they will be able to browse the messages in the digest on an individual basis.

Using mailing lists - commands and methods

See also: Other mail server commands

OK, you've created your mailing list... Now what do you do with it?

In general, the answer to this question depends on whether the list you have created is *moderated* or *unmoderated*. For moderated lists, only users marked as list moderators may send messages to the list: other users, even members, cannot send mail directly to the list. Moderated lists are useful for low-volume announcement lists, or in cases where the subject matter sent to the list needs to be scrutinized before posting.

In the normal case, however, the list will be unmoderated, which means that your users can manage their own subscriptions to it by sending commands to the Mercury Mail Server via e-mail.

Commands affecting list membership or operation should be sent to the mail server address: by default, this will be the reserved address **`maiser`** at your site. Maiser is not a username - it is a kind of alias handled in a special way by Mercury itself. Sending a message to maiser tells Mercury that the message body contains commands that it needs to process, rather than mail that needs to be delivered. Multiple commands can be included in a single message, one line per complete command, and command processing terminates as soon as Mercury encounters a blank line or an `EXIT` command. The user who sends the message will receive a short message back indicating the success or failure of the commands he has issued.

For mailing list management, the following commands are recognized by the mail server:

Commands available to everyone

SUBSCRIBE <list-name> Add the sender's address to the list
(also *SUB <list-name>*)

UNSUBSCRIBE <list-name> Remove the sender's address from the list
(also *UNSUB <list-name>*,
or *SIGNOFF <list-name>)*

ENUMERATE <list-name> Return the list membership
(also *REVIEW <list-name>)*

LIST Returns the lists available at this host

SET <list-name> DIGEST Set your list subscription to digest mode

SET <list-name> NODIGEST Turn off digest mode for a list.

SET <list-name> MAIL Turn on delivery from a list

SET <list-name> NOMAIL Turn off delivery from a list

STATUS <list-name> Get current subscription information for a list

Commands only available to list moderators:

ADD <list-name> <address> Add a user to a list

REMOVE <list-name> <address> Remove a user from a list

MSET <user> <list> <option> Change a user's subscription options
- *option* can be MAIL, NOMAIL, DIGEST or NODIGEST

MSTATUS <list-name> <user> Get a user's subscription status

PASSWORD <password> Supply the password for moderator commands

For more information on passwords and lists, click here.

Other mail server commands

Other mail server commands

As well as providing automated control of mailing lists, Mercury's mail server also recognizes the following commands; to use these commands, send a message to the Mail Server account (usually called "maiser") with the message body containing the commands you want it to execute, one per line. The Mail Server will process commands until it encounters a blank line or an EXIT command.

HELP Returns the helpfile defined in the MAISER section of MERCURY.INI to the sender.

BOUNCE Returns the message to the sender, headers intact.

LOOKUP <string> Searches the local system for user names matching <string>, which can contain '*' and '?' wildcards. If you wish to disable lookup, make sure that there is no LOOKUPFILE entry in the MAISER section of your MERCURY.INI.

VERIFY <address> Returns a message indicating whether the address specified is valid on the local host.

INDEX Sends the file INDEX.TXT from your "files to send" directory (identical to the command *SEND INDEX.TXT*).

SEND <filename> Sends the named text file from your "files to send" directory. If you do NOT want this facility to be available on your system, make sure that there is no SEND_DIR entry in the MAISER section of MERCURY.INI. The send command is VERY secure - you cannot specify paths of any kind in <filename>, and Mercury will ONLY look in the directory specified in SEND_DIR.

FINGER <address> Returns information about the address supplied. Also returns the contents of the file *PROFILE* if it exists in the specified user's new mail directory (useful for PGP keys and the like).

EXIT Stop processing commands from the mail message.

Configuring Template Files

Template files are files used by Mercury to generate messages automatically. In a template file, you can enter plain text, and also special substitution characters that Mercury will replace with system-specific information. Delivery failure notifications, confirmations of delivery and some of the mail server responses are formatted using template files.

A template file is a plain text file and can be created using any standard editor, for example the Windows NOTEPAD command. It must be formatted as a mail message - in fact, the first four lines of the message will usually look like this:

```
From: postmaster@~N (Mail System Administrator)
To: ~T
Subject: Confirmation of delivery
Date: ~D
```

The ~N, ~T and ~D characters are special substitutions, replaced with the system's domain name, the recipient's mail address and the date respectively. The rest of the message can take any form you wish and you can use any of the special substitutions as often as you need.

Mercury recognizes the following substitutions in template files:

~~	A single tilde character
~D	The date, in proper RFC822 format
~T	The recipient's mail address
~G	The first x lines of the message (*)
~M	The entire original message
~B	The entire original message body (no headers)
~R	The failure text, or results (for mail server searches)
~S	The subject field from the original message
~N	The current system's Internet domain name
~Y	A valid MIME Multipart boundary separator

Using Multipart MIME format in Template files: MIME is the dominant Internet standard for message formatting. One of the more powerful features of MIME is its ability to generate messages with multiple parts: in order to do this, you need to add some special headers to the message, and to separate the parts of the message from each other using a special boundary string. To generate a Multipart MIME message in a template file, add the following two lines to the headers of your template file, exactly as they are shown:

```
MIME-Version: 1.0
Content-type: Multipart/Mixed; boundary=~Y
```

Now, at the start of each part of your message, add the following lines

```
--~Y
Content-type: Text/Plain
```

Making sure that there is a single blank line between these lines and the start of the text of the message part.

For an example of how to generate Multipart MIME messages in Mercury, please see the sample delivery failure template file, FAILURE.MER, supplied with Mercury.

* The number of lines copied from the original message is controlled by the option settable in the *Mercury*

core module configuration dialog.

Configuring the Mail Server

See also: Mailing list commands and Other mail server commands

Mercury's Mail Server provides a number of automated services, including automatic mailing subscription and unsubscription, file transmission, user lookup and search facilities, and remailing files at specific times.

General mail server configuration:

Help file The file Mercury should send when it receives a "help" command, or when it receives a command it does not recognize.

Lookup results file The name of a template file that the mail server should use to return the results of user searches using the *Lookup* command. If this field is left blank, then the lookup command will be disabled.

Log file The name of a file in which the mail server should record all the commands it processes. If this field is left blank, the mail server will not perform any logging.

"Send" directory The directory in which the mail server should search for files requested using the *Send* command. Files in this directory must be text files, so if you want to make binary files available via the Send command, you will have to uuencode them yourself first. The mail server will only look in the directory you specify here and will not accept filenames containing paths; because of this, the option is an extremely safe way of distributing data to the public via e-mail. If this entry is blank, then the Send command will be disabled.

"Notify" queue directory Mercury's mail server supports two deferred mail commands - *Notify*, which sends a broadcast message to the sender at a given time (if broadcasts are supported on your network), and *Remail*, which sends a mail message at a particular time. For these commands to be available, Mercury requires a directory where it can create status files for each request: enter the path to that directory in this field (the directory must exist already - Mercury will not create it). If this field is blank, the "notify" and "remail" commands will be unavailable.

Disable the mail server "Lookup" command Check this control if you do not want the Lookup command to be available on your system.

Only accept "notify" commands from local users If this command is checked, then the mail server will only accept "notify" requests from users who are local to your system (that is, to whom it could actually deliver a message). If the control is unchecked, then anyone, no matter where they are located, may queue "notify" requests for users on your server.

Editing the mail server template files

The options to edit the mail server help file and to edit the lookup results template let you customise the responses generated by the mail server to certain commands. The help file is a plain text file - template substitutions cannot be used in it.

Configuring support for Groups

See also: Core module configuration

If your underlying Network system supports the notion of "Groups", or collections of users, then you can enable that support using this configuration dialog.

By default, Mercury does not make groups available for mailing purposes - this is partially a security issue and partially a configurability issue. In order to make a group on your system available to receive mail, you must add it here. Making a group available involves providing three pieces of information:

Public name The *public name* of a group is the e-mail address people will use to send mail to the group. You can give a group the same public name as its actual name on your system, but there may often be reasons why you might not want to do this - for instance, you might feel that the group "everyone" on your Novell NetWare server is less suitable than the name "staff", so you would define the group's public name to be "staff". People would then mail everyone on your server by sending a message to staff@server.domain. You will also need to use different public names for groups on different servers that have the same group name.

Group name The actual name of the group on your network. The group's public name may be different from this name.

Host name The server or host on which the group is based. In single-server environments you will not have to enter anything in this field, and the value entered here will vary depending on the underlying network: for instance, under Novell NetWare Bindery Mode, the host name will be the name of the Bindery Server that holds this group.

Example:

Your NetWare server's Internet name is "orange.com", and you have a group on it called SUPPORT, which you want people to be able to mail as "tech-support@orange.com".

In the Public name field enter	tech-support
In the Group name field enter	SUPPORT

Note that when defining groups you do NOT add the domain name.

Configuring aliases, synonyms and autoresponders

An *alias* is a specialised form of e-mail address that stands in for another e-mail address on your system. Aliases are often used to create addresses which do not vary, even though the person receiving the mail may change. For example, say you want to offer your users an e-mail address they can use to obtain help; it is clearly much better to use an address like help@mydomain.com than to give the address of a user on your system, since if that user leaves or is transferred, you can simply point the alias for "help" at the person's replacement and your users are not forced to change their habits.

A *synonym*is a specialised form of alias that effectively replaces a user's real e-mail address with an alternative form. Using synonyms, you can adopt consistent addressing schemes that differ from your users' actual login names. For more information on creating and using synonyms, click here.

Mercury has very powerful aliasing features: you can access them either from this dialog, or by using the commandline import/export tool MALIAS.EXE supplied with Mercury. An alias simply consists of two parts - the alias (or, the address people use to send mail) and the real world address (the address to which Mercury should deliver any messages it receives addressed to the alias). The real world address does not have to be a local address - it is perfectly valid to have an alias for an address on a remote system (this approach is often used to redirect mail to someone while they are absent, or if they leave the organization).

To create an alias, fill in the alias and the real world address, then click the *Add as new* button.

To change either the alias or the real world address of an existing alias, click on it in the list, then make the changes and press the *Change* button.

To remove an alias, click on it in the list then press the *Delete* button.

Exporting aliases: You can save your alias list to a simple text file in the format expected by the MALIAS commandline utility by clicking the *Export* button.

Autoresponders

An autoresponder is an e-mail address that simply returns a predefined message to whomever sends it mail. Autoresponders differ from automatic replies in that once the automatic response has been sent, no further attempt is made to deliver the incoming message.

In order to support autoresponders, Mercury supports two specialised aliases that work differently from other aliases - *FILE:* aliases and *TFILE:* aliases. A *FILE:* alias is an address that will return the contents of a text file to the sender when it receives any message, while a *TFILE:* alias returns a formatted message using a template file.

To create a *FILE:* or *TFILE:* alias, enter the alias as normal, but for the real world address enter either TFILE: or FILE: followed immediately by the path to the file you want to use.

Example:

```
        info   =   FILE:\\myserver\sys\system\mercury\info.txt
or      faqs   =   TFILE:r:\system\mercury\faq.mer
```

Note that it is very important that the file specified in a TFILE: alias is actually a template file: if you do not specify a valid template file, Mercury may crash when it tries to send the reply.

TFILE: and FILE: aliases are completely secure - they are only accepted if they actually appear in your alias file: a user cannot send a message to a TFILE: address to obtain files illegally from your system.

Configuring file locations and Queue settings

See also: Core module configuration

This dialog tells Mercury where to find various files that it uses in regular operation. There should usually be little or no need to change these values. Note that when you are using Mercury on a Network, all paths should be entered in *UNC format* - like this: *\\SERVER\VOLUME\PATH*. It is permissible to use DOS paths as well, but you should not use non-standard paths, such as the Novell NetWare path format. In all cases except for the log file entry it is permissible for the file not to exist - Mercury will create it as required.

"List of lists" file The location and name of the file in which Pegasus Mail stores information about the mailing lists available on your system.

Alias database file The location and name of the file in which your system aliases are stored.

Synonym database file Synonyms are a specialised form of alias used in conjunction with the Pegasus Mail system to provide alternative addressing formats on your system. If you have created synonyms on your system using the Pegasus Mail PMGRANT utility, you will need to use the CH_SYN.EXE utility (or NSYNONYM.EXE in NetWare NDS mode) supplied with Mercury to build a synonym database for Mercury, and enter the name and location of that file here. Note that the Pegasus Mail NDS-mode configuration utility, NCONFIG.EXE, can also be used to create synonyms for your users - this utility is a standard part of Pegasus Mail for Windows releases starting with v3.12. For more information on synonyms, click here.

Delivery confirmation template The name and location of a template file that Mercury should use when reporting confirmation of delivery.

Delivery failure template The name and location of a template file that Mercury should use when reporting delivery failures.

System log file The name and location of a file into which Mercury should store information about the jobs it processes. If this entry is left empty, Mercury will not perform any logging.

Directory for noticeboards If you have created a noticeboard system within Pegasus Mail, Mercury can delivery mail to it. Enter the top directory in your noticeboard structure here (exactly the same as the NB environment variable you give to Pegasus Mail). Mail can be sent to any noticeboard using the address format *<boardname>%nb@host.domain* - so, for example, if you have a noticeboard called *comp.sys.mail*, you could mail it using the address *comp.sys.mail%nb@host.domain*. *Tip:* you can make the process of posting to noticeboards much easier by creating aliases for them - using an alias allows you to create a simple e-mail address for a noticeboard.

Queue Processing controls

These two settings control how frequently Mercury/32 should retry messages that have temporary delivery problems, and the maximum number of times a job should be retried before Mercury should conclude that it cannot be delivered. You cannot set the retry period shorter than one minute, nor can you set fewer than two retry attempts.

Process the queue in two passes (file locking) If you check this control, Mercury will change the way it processes mail submitted by programs such as Pegasus Mail; instead of taking the submitted job immediately, it will wait until the job has the same non-zero size for two polling cycles in a row before processing it. This can be necessary in systems such as Windows Peer-to-Peer networking where file locking is not properly implemented; it ensures that the client has finished writing the mail message to the queue before Mercury tries to process it. Turning this option on is always safe, but will result in a slightly

longer delay in mail being sent out.

Entering user information

When you are using Mercury/32 in an environment for which it has no network support module, the option to create and manage local users becomes available on the Configuration menu. This option allows you to manage the Pegasus Mail user configuration file PMAIL.USR, which is used by both Mercury/32 and Pegasus Mail to define and locate the users who exist on the system.

Important note: Mercury reads the Pegasus Mail system file PMAIL.USR to obtain local user information, and caches it in memory while it runs. If you make changes to PMAIL.USR using Pegasus Mail or via any other process that manipulates the file directly, you will need open the Mercury user administration dialog while holding down the CTRL key, to force it to re-scan the list. Restarting Mercury will also work in this situation. Until Mercury refreshes its cached copy of the user list, it will not be able to deliver to users added via the alternative method. Mercury saves changes made by its own user dialog as soon as you click OK, so we recommend that you do all your user administration from within Mercury.

Username the name the user will give to identify himself to Pegasus Mail. By default, this is also the user's address. In order to remain compatible with the DOS and Win16 versions of Pegasus Mail, usernames are limited to eight characters total length (because they are used as the name of a directory).

Personal name the "familiar name" by which the user is known to the rest of the world. This is the name that appears in the Pegasus Mail *Local user list* function, and which Mercury uses when responding to mail server VERIFY and LOOKUP commands.

POP3 password The password the user needs to enter when retrieving mail via POP3. If a user does not have a POP3 password, or has a blank POP3 password, he or she cannot retrieve mail from Mercury/32 using the POP3 protocol. The password can be from 1 to 48 characters in length.

APOP secret APOP is a specialised POP3 protocol command that provides a reasonably secure form of login without the user having to send his or her password in clear text over the network. APOP is based on the idea of a shared secret - some piece of text known both to the user and to Mercury. If the user has a mail program that supports the APOP command, enter the shared secret in this field. The shared secret should be at least 8 characters long and can be any length up to 48 characters. It can be the same as the user's POP3 password, but need not be (and in fact, we do not normally recommend that it be so).

In networked mode (where Mercury is using a network interface module such as its Novell NetWare modules) the POP3 password and APOP secret will be obtained or defined in a different way - see the documentation provided with the module for more information.

Administrator privileges A user with administrator privileges can create, edit, rename and delete users on the system from within Pegasus Mail.

Copy default messages This control is only available when you are creating a user - it is disabled if you are editing a user's information. If checked, it tells Mercury to copy any file with the extension .DMI in the directory where MERCURY.EXE is installed into the new user's new mail directory. .DMI files (*Default Message Interface*) should be properly-formatted RFC822 mail messages and will appear in the user's new mail folder the first time he or she runs Pegasus Mail. They are normally used to provide information on using the system, or welcome information.

Configuring Pegasus Mail to work with Mercury/32

While Mercury/32 can be used with a variety of mail programs, it is primarily intended for use with Pegasus Mail, and has a special option to configure any copy of Pegasus Mail to work with it. To use this dialog, enter the full path to the directory where the Pegasus Mail .EXE file you want to configure is located. Mercury will create the files necessary to allow Pegasus Mail to submit mail to Mercury for processing.

Mail Submission (queue) directory Enter here the path your copies of Pegasus Mail should use to find the directory where they should place mail for processing by Mercury. The default value is Mercury's own queue directory, but this may not be suitable for, or accessible by other workstations. As an example: say Mercury's queue directory is C:\QUEUE; in order to make that queue available to other users on the network, you need to set up a share - but you can't share it as C:, because all systems will already have a C: drive. Instead, you set up the share as T:. In this scenario, Mercury needs to access C:\QUEUE, but the Pegasus Mail clients need to submit mail in T:\QUEUE: the directory is the same, but the path specification is different.

If you have copies of Pegasus Mail (for instance, DOS and Windows versions) in different directories, you will need to use this option once for each version you have installed. At present, the Macintosh version of Pegasus Mail cannot interact with Mercury other than in native Novell NetWare mode.

Sending a mail message from within Mercury

Mercury/32 includes an option you can use to compose and send simple mail messages. The facility is rudimentary, and is not intended to replace a full-fledged mail client such as Pegasus Mail. To send a mail message, choose *Send message* from the *File* menu. Mercury will open the message editor dialog with the "From field" entry filled with the postmaster address for your system. Enter the recipient of the message in the *To:* field (you may enter more than one address, provided they are separated by commas), the subject of the message and the message body, then click the *Send* button.

Tip: If you want to use this option to send a mail message to a Mercury mailing list, make sure you change the From field of the message to something other than the postmaster address - Mercury's mailing list manager contains special traps to prevent mail storms, and one of those traps will prevent mail from the postmaster account being distributed to lists.

Urgent sets the urgent flag in the message. Different mail clients will respond to this flag in different ways - Pegasus Mail, for instance, will display urgent messages in red at the top of the new mail list. This option does *not*, however, make the message travel any faster.

Confirm delivery sets a flag in the message asking for a confirmation from the recipient's mail system that the message has been delivered to his mailbox. Getting a confirmation of delivery does not mean that the recipient has read the message - only that it has been successfully delivered and is available for him to read. Many, but not all, mail systems on the Internet will provide confirmation of delivery. The confirmation will be sent to whatever address appears in the *From field* of the message editor dialog.

Confirm reading Sets a flag in the message that asks the recipient's mail program to confirm that the message has actually been read. At the time of writing, only Pegasus Mail supports this flag, and on some systems the user may be able to decline the confirmation (many people regard such confirmations as an invasion of privacy). The confirmation will be sent to whatever address appears in the *From field* of the message editor dialog.

You cannot attach files to messages sent using this facility, and the total size of the message may not exceed 30,000 characters.

Mercury/32 License

Mercury/32 is fully proprietary copyrighted software. David Harris ("The Author"), asserts his right to be recognized as the author and copyright owner under International Treaty and Copyright Law. The reservation of these rights notwithstanding, the Author wishes to provide the Mercury Software freely for the benefit of the broader Internet User Community, and especially for the benefit of users of his Pegasus Mail mail client.

You are granted a non-exclusive license to use the Mercury/32 software on as many computers as you wish subject to the following terms and conditions:

Consideration: You may use the Mercury/32 software without any fee or charge, although you may at your option choose to license manuals as described elsewhere in this guide. You agree that the author of the Mercury/32 system receives consideration from your use of the program in the form of acclaim and valuable reputation, and that this consideration is sufficient to bind you to the terms of this license even if you opt not to purchase manuals.

Prohibited uses: Mercury/32 may not be used as a transport for the dissemination of Bulk Unsolicited Commercial E-mail(commonly referred to as "spam"). For the purposes of this section, *Bulk Unsolicited Commercial E-mail* shall be taken to mean e-mail containing advertisements for products or services, which is sent to lists of more than 50 addresses where the addressees have not explicitly expressed interest in receiving such e-mail.

No transfer of rights: You agree that your use of the Mercury software entails no transfer of intellectual property rights, copyright or ownership from the Author.

No liability: The Mercury software is provided without warranty of performance. While the author has taken every care to ensure that the software works correctly and without fault, you agree to hold the Author blameless for any and all damage or loss caused by its use.

Distribution and supply: You may distribute the Mercury/32 Software to any person or organization by whatever means you wish provided that:

a) You do not charge more than reasonable handling or media production fees for the supply of the Mercury/32 Software itself.
b) You do not represent ownership or title over the Mercury/32 Software.
c) You supply the Mercury/32 Software complete and unmodified.
d) You do not supply the Mercury/32 software as part of, or in conjunction with, any package or promotion explicitly designed for the purpose of disseminating Bulk Unsolicited Commercial E-mail.

Note: the author explicitly permits commercial distribution of the Mercury Software as part of Commercial Software packages provided the conditions of this license are fulfilled and the Author's written permission is obtained prior to shipping.

Setting up Automatic Replies

Mercury can manage auto replies for Internet mail, much the same way the UNIX "Vacation" program does. To enable auto-reply for a user, create a file in his/her new mail subdirectory called AREPLY.PM, containing the message to be returned to the sender. Mercury will look for this when delivering mail to the account, and will return its contents immediately the message is delivered locally.

To avoid the possibility of mail storms (which can happen when two accounts start auto-replying to each other, or when an autoreply is sent to a list server), Mercury remembers every address from which a message has been successfully delivered for the account in the last 48 hours; if more than one message comes in from the same address in any 48 hour period, Mercury will generate only one auto-reply. The auto-reply memory is stored in a file called AREPLY.KFL in the user's new mail directory.

You can also specify a static kill file for autoreplies on a user by user basis. When generating an autoreply, Mercury checks to see if there is a file called AREPLY.KFS in the user's mail directory. If there is, it is checked for the address before the autoreply is transmitted. AREPLY.KFS differs from the AREPLY.KFL file Mercury generates automatically in that Mercury never changes or deletes it. It is provided to allow a user to suppress certain autoreplies permanently. Addresses should be entered into AREPLY.KFS one per line, in their simplest form.

Manuals

Mercury is free software, but to support its development, I make manuals available for sale. Two types of manual are available - single licenses, and licenses which allow you to make as many copies as you wish for use at your site. Purchasing any manual license also allows you to contact us directly for personal technical support.

A single copy license gets you one bound copy of the Mercury Manual and the most recent version of the program on CD-ROM. You may not make make further copies of the manual, other than the single copy for archival purposes that is permitted by copyright law. Single copy licenses cost US$150.

Site licenses get you a single master copy of the Mercury manual and a license that permits you to duplicate the master copy as many times as you wish for use at your site. As with Pegasus Mail, a "site" is considered to be the extent of your organization in one city. Site licenses cost US$300, and again include a current version CD-ROM.

> *If you already hold a site license for any version of Pegasus Mail, you may claim a US$75 deduction from the price of a Mercury Site license. No deduction applies to single-copy license purchases, or if you hold a 10-copy or single-copy Pegasus Mail license.*

Remember that manual purchases are optional - there is no obligation for you to purchase manuals for Mercury, but if you do so, you are making a positive contribution to its development and helping me to support it.

How to order manuals
Payment by credit card
Terms and conditions of orders
Order form for Mercury Manuals

How to order manuals

To order a Mercury Manual, please print and fill out the order form (a copy of the order form is also supplied in ASCII format as ORDER.FRM). Then send it with your cheque or purchase order to:

Pegasus Mail, c/- David Harris,
P.O. Box 5451,
Dunedin,
New Zealand. **(or Fax +64 3 453 6612)**

Company purchase order forms are accepted from any educational or charitable institution or government department, and from companies with a paid up capital in excess of $30,000.

Payment must be made in ***US dollars, drawn on a US bank***. There is usually a delay of around 7-14 days before a manual can be sent out, and we reserve the right to hold your order if a new release is pending (so you can get the new release). Manuals are always sent by courier (2-4 day delivery anywhere in the world); the cost of freight can be calculated using the freight table.

All manual orders are shipped with a diskette containing the most recent version of the program.

PLEASE, PLEASE! Remember to enclose the order form: I'm not psychic, and I can't guess at delivery addresses from a cheque alone...

Payment by credit card

We can accept payment for Manual orders on any valid VISA or Mastercard. If you wish to pay this way, please note your card number, expiry date, and the name on the card on the order form, and make sure you sign our order form. We cannot accept Diners Club or American Express.

Special Conditions:

* The amount you are billed will probably not be exactly the amount on the order, because we have to submit the voucher to the VISA Centre converted to New Zealand dollars. We will use the USD conversion rate that applies on the day your order is processed.

Freight costs for shipping manuals

Manuals are shipped via courier to any location in the world. The delivery time is usually 3-4 days and is fully traced. Freight costs are as follows:

When ordering one manual, add:

To Australia and New Zealand	US$15.00
To the U.S.A. and Canada	US$27.50
To the U.K. and Europe	US$32.50
All other destinations	US$37.50

When ordering two or more manuals, or any Pegasus Mail Bundle option, add:

To Australia and New Zealand	US$20.00
To the U.S.A. and Canada	US$34.00
To the U.K. and Europe	US$39.00
All other destinations	US$45.00

Examples:

* Shipping a Mercury site license to Sydney, Australia, add US$15
* Shipping a DOS and Windows bundle to Cincinatti, U.S.A, add US$34
* Shipping four Mercury Single-copy manuals to Germany, add US$39.00

Terms and conditions applying to orders

The following terms and conditions apply to all orders for Mercury MTS manual sets. Submitting an order indicates acceptance of these terms and conditions in their entirety, over and above any waivers, disclaimers or conditions stated on the order itself.

Site license A site license for Mercury MTS manuals provides you with one master copy of the manual and a license permitting you to make an unlimited number of copies of that master for use at one site. A "site" is defined as the extent of an organization in one metropolitan area, so if you have four offices in one town, one site license will cover them all. Only ONE organization may hold a site license - you may not distribute copies of a site-licensed master outside your organization. Note that State or National Governments and Local bodies do not consitute organizations for the purposes of this definition. Site licensed manuals include ASCII and RTF versions of the manual on diskette.

Single copy license You receive a single master copy of the manual, and a license permitting you to make exactly one copy of that original for archival purposes only.

Media *A*ll manual licenses include a CD-ROM with the most recent version of the Software.

Credit terms Our terms for payment on credit are STRICTLY net 30 days. No monthly statement is issued - we require payment on invoice.

Deductions We require that the total amount of our invoice be paid to us; it appears to be "standard operating procedure" for some banks to deduct fees before sending payment to us, which we will not accept. Please double-check that your bank understands that no deductions must be made from the amount sent to us or your payment will be rejected.

Freight charges Manuals are always shipped via courier shipment anywhere in the world, and the freight cost is extra. Our courier usually provides a four-day service to any destination. We can ship via airmail on specific request, but are unable to ship via other carriers (such as FedEx, UPS or TNT). Freight prices can be calculated using the freight table.

Prepayment If you prepay your order, your payment will not be presented until the day your manual is shipped. In general, we prefer that you do not prepay orders - we would sooner receive a purchase order and accept payment on invoice.

Pro-forma invoices, acknowledgments, quotations and bids Because of the sheer volume of paperwork we have to process, we regret that we cannot supply pro-forma invoices, quotations or bids, and cannot acknowledge receipt of orders other than via e-mail on specific request only.

US IRS taxpayer requirements The vendor is not an American citizen and is not resident in the United States for tax purposes: consequently, requests for FEINs, social security numbers or other US Taxation requirements cannot be complied with and will receive no reply.

Withholding taxes on payments We will not accept payments from which withholding tax deductions have been made unless those deductions have been made in compliance with the laws of New Zealand. If you deduct withholding tax from your remittance, your payment will be rejected even if you do so in compliance with the laws of your own country. Such deductions are a matter between you and your own government.

Credit card payments Credit card payments will be converted to New Zealand dollars using the prevailing exchange rate on the date your manual is shipped to you. The amount you see on your credit card statement will probably not be exactly the amount shown on the order form because of currency conversions.

Please read Terms and Conditions carefully before placing your order.

```
        +-------------------------------------------------+
        |         O R D E R   F O R M,   v 3 . 0 1        |
        |              for Mercury/32 Manuals             |
        |     Order form valid until December 1st 2000    |
        +-------------------------------------------------+
 Ship to:                          Bill to:
[                            ]    [                               ]
[                            ]    [                               ]
[                            ]    [                               ]
[                            ]    [                               ]
[                            ]    [                               ]

Date this order was mailed          [   ] Day  [   ] Mon  [   ] Year
What is your purchase order number  [                              ]
 or: Credit card type (tick one)    [   ] Visa [   ] Mastercard
     Card number (print carefully)  [                    ] exp: [   ]
     Name shown on card             [                              ]
Number of invoice copies required   [   ]
Internet e-mail address of contact  [                              ]
Fax number of contact               [                              ]
```

Order status and update information cannot be sent if you do not provide either a contact e-mail address (preferred) or a fax number. If paying by credit card, you must sign the copy of this order form you send to us.

Please supply the following items:

[ ] Single manual license for Mercury @ US$150 . . . . US$ [ ]

[ ] Site-licensed manual for Mercury @ US$300 . . . . US$ [ ]

Freight cost calculated using freight list US$ [ ]

[ ] We already hold a Pegasus Mail SITE license - please give us a US$75 deduction from a Mercury ->SITE<- license purchase! (Discounts do NOT apply to single-copy license purchases).

Ordering instructions: Please complete this form, and send it via airmail with your cheque or purchase order to:

Pegasus Mail, c/- David Harris,
P.O. Box 5451, Dunedin,
New Zealand.

or fax it and your purchase order number to: (+64) 3 453-6612.

Please allow 7-14 days for your order to be processed.

Configuring Statistics and System Messages

See also: Core module configuration

Mercury keeps extensive statistics about its operation, which can be useful for working out loading and peak traffic times, and for tracking down performance bottlenecks. It also supports a System Messages window, which acts as a kind of "console" in which the various co-operating parts of the Mercury system can report events and messages.

Statistics

Save statistics to a file periodically Checking this control tells Mercury to save a "snapshot" of the current statistics tree to a file in the directory you specify at the frequency you specify. *Save in* should be filled out with the name of a directory (***not*** a filename) in which Mercury should create its snapshot files. Snapshot files have a name representing the date and hour they were saved, and the extension .MSR.

E-mail statistics periodically Checking this control tells Mercury that it should e-mail a "snapshot" of the current statistics tree to a specified address (which does not have to be local) at the frequency you specify. *Send to* should be filled out with any single e-mail address.

Automatically open the statistics window at startup If this control is checked, the statistics window will be opened automatically every time you run Mercury.

Collect statistics about mail sent by local users When this control is checked, Mercury will record the size and number of mail messages sent by each local user on your system. This can consume quite a lot of memory, particularly if you have many users, but it's an extremely useful way of seeing who is using your system's resources.

System Messages

System message reporting level This controls the verbosity of the information reported in the System Messages window. Level 0 disables reporting altogether, while level 5 is typically used to report low-level program status and debugging information. There is usually no reason to adjust this control from its default level 3 setting. At level 3, Mercury reports useful information about delivery problems and actions within your system.

Number of messages to store Controls the maximum number of items that will appear in the System Messages window. Once the total number of items in the window exceeds the number you specify, the earliest entries in the list will be successively discarded. The larger you set this number, the more memory will be used by the System Messages window, although the amounts are not enormous even at quite high numbers, like 1000 lines.

Automatically open the System Messages window at startup If this control is checked, the System Messages window will be opened automatically every time you run Mercury.

MercuryS, the Mercury/32 SMTP Server

MercuryS is the Mercury SMTP Server Module. It listens for incoming SMTP connections from the outside world, placing incoming mail in the Mercury spool directory for processing when it receives them.

SMTP (Simple Mail Transfer Protocol) is described in detail in Internet Standards Documents RFC821 and RFC1652, which are available via FTP from nic.ddn.mil in /rfc, as well as from various other places on the Internet.

Configuring MercuryS

Configuring the Mercury SMTP Server module

See also: Logging and session logging
Connection control and relaying

Announce myself as In some situations, you may wish to have your SMTP server to tell clients connecting to it that its name is something other than the value in the Mercury *My Name* field. An example of a situation when this might be necessary is when your *My Name* field represents an entire domain for which Mercury is acting, but you want it to identify itself to connecting clients using its real Internet machine name. In the majority of cases this field can and should be left blank.

TCP/IP Timeout the length of time in seconds that MercuryS should wait for data on a connection before assuming that the connection is no longer valid and aborting it.

ESMTP maximum size If non-zero, the maximum size message MercuryS should accept from compliant ESMTP clients. MercuryS will advertise this via the ESMTP SIZE keyword. Not all clients, even ESMTP clients, will honour this setting.

Listen on TCP/IP port By default, MercuryS listens for connections from the outside world on port 25, which is the standard reserved port for the SMTP protocol. In some cases, particularly when you are behind a firewall, you may wish to listen on an alternative port - enter the number of that port in this field. **If you change this field and save the dialog, you will need to exit and restart Mercury/32 before the change will take effect**.

IP Interface to use If your computer supports multiple IP interfaces, you can use this field to tell MercuryS which interface it should select when listening for connections: enter the interface as a dotted IP address in the general form *www.xxx.yyy.zzz*. As an example, your system may have one IP address assigned to a dialup PPP connection, and another, different IP address assigned to a local Ethernet network - you would enter here the interface MercuryS should use. If you leave this field blank, MercuryS will listen on all available interfaces. Unless you are *very* sure of what you are doing, or have been instructed by an ISP or network administrator, you should leave this field blank. If you change the IP interface in this field, you must restart Mercury before the new interface number will be used.

Sender kill file MercuryS allows you to create a file of addresses from which you will refuse to accept mail. The file can restrict individual addresses, or (using wildcard characters) entire domains or groups of users. This feature can be useful for dealing with spam, or with abusive correspondents. When a message is "killed" by the killfile, you don't even receive the data, so it is an excellent way of protecting yourself from denial of service attacks. Be careful, though - once someone is in your MercuryS killfile, they cannot send you mail at all - you will need to work out for yourself whether or not this presents any problems. To edit the contents of your killfile, click the *Edit* button next to the field. **Note:* using a killfile with more than a few entries can impact significantly on the speed of incoming mail processing.

Display session debugging information Check this control if you want the MercuryS console screen to display more verbose information about each connection as it comes in.

Accept 8BITMIME data connections If this control is checked, Mercury will tell connecting clients that it supports the 8BITMIME SMTP extension. What this means is that Mercury will tell connecting systems that it can handle mail messages containing 8-bit data, bypassing the normal 7-bit restriction on Internet Mail data. It is very important to note that Mercury currently cannot convert 8-bit data to 7-bit data when it passes it on to other SMTP systems, as is required by the 8BITMIME specification: in practice, this is unlikely to cause problems in the majority of cases, but you should be aware that enabling this control has the potential to produce undesirable effects in rare instances.

Accept mail for invalid local addresses In regular use, MercuryS will refuse to accept any message that appears to be addressed to a local user who does not exist. This refusal can result in the sender getting

unhelpful mail messages from their mail program. If you check this control, Mercury will accept the message and the Mercury core module will later reject it and send it back to the sender, but in a more clearly-explained form. Mercury will also refer a copy to the postmaster, who can then correct any addressing error and pass it on to the proper recipient.

MercuryS: Connection and relaying controls

See also: Configuring MercuryS

Connection control

The Connection Control section allows you to place restrictions on the hosts from which MercuryS will accept connections. To add an entry to the list, click the *Add restriction* button then type the IP address in the *IP Address* field and select either *Allow* or *Refuse*, and click *OK*. The digit 0 acts as a wildcard in a connection control entry, so adding an entry refusing access to 165.25.9.0 will cause MercuryS to refuse connections from any machine whose address's first three octets are 165.25.9. Note that there is an implicit rule "Allow 0.0.0.0" at the end of this list, so if an address "drops through" the list, it will be automatically accepted.

To edit a connection control entry, highlight it in the list, then click the *Change selection* button.

Controlling relaying

SMTP relaying is the standard method of propagating mail on the Internet: in normal operation, an SMTP host will accept any message destined for any user, even if that user is not a local user on the system: after it has accepted the message, it will *relay* it to the correct host for delivery. Mail agents like Pegasus Mail and Eudora routinely depend on relaying to send mail.

In recent times, relaying has been abused by perpetrators of mass unsolicited commercial e-mail (or "spam"), and many sites wish to control the way relaying is managed. Mercury provides two anti-relaying modes, *normal* and *strict*. Normal mode is turned on by checking the control labelled "*Do not permit SMTP relaying of non-local mail*". Strict mode is turned on by also checking the control labelled "*Use strict local relaying restrictions*". The default for these controls is both off.

In either mode, Mercury will ***always*** accept mail addressed to any local address. Similarly, mail to any address for which Mercury holds an alias will also be accepted, even if the alias resolves to a non-local address.

In normal anti-relaying mode, Mercury will accept mail for delivery if either the recipient or the originator has a local e-mail address. If neither address is local, Mercury will compare the IP address of the connecting host to its connection control list (see above): if it finds an "allow" entry in that list that explicitly includes the connecting machine, then it will accept the mail, otherwise it will be failed with the diagnostic "571 - Sorry, we do not relay non-local mail".

In strict anti-relaying mode, Mercury follows the normal rules described above, but if the "From" address appears to be local, then Mercury will search the connection control list and will only accept the mail if an "allow" entry appears that explicitly permits the connecting host.

The difference between the two modes is that normal mode requires less setup and maintenance, but is less secure, while strict mode practically guarantees that no unauthorised relaying can occur at the expense of having to manage a list of permitted relay hosts.

When you configure Mercury to operate in strict mode, you must ensure that you add "allow" entries to your connection control list for every machine that is to be permitted to relay mail via this copy of Mercury. Note that this does NOT mean that you have to enter the address of every machine from which you want to accept mail - mail to local recipients is always accepted, regardless of the relaying mode. Strict mode only requires "allow" entries for machines from which Mercury is to accept mail to be delivered to non-local addresses.

Anti-relaying is turned off by default in Mercury, because relaying is the proper, defined action under the SMTP protocol. It is, however, almost always safe to turn on normal anti-relaying mode.

Authenticated SMTP

Mercury supports a recent Internet standard called Authenticated SMTP: when this feature is enabled, Mercury will advertise to connecting clients that it can accept SMTP authentication. If a client then authenticates correctly, it will be allowed to relay. Pegasus Mail v3.12 and other widely-used Internet mail clients support authenticated SMTP, and it is an excellent way of allowing your roving users to use your server without opening yourself to relay abuse.

Authenticated SMTP requires that both the client and server have access to a common password. For that reason, you need to provide Mercury with a list of usernames and the passwords that correspond to them - Mercury typically cannot get this information from the operating system. Enter the name of the file where Mercury should store the user/password combinations, then click the *Edit* button to edit it. Each line contains one username/password pair.

MercuryC, SMTP Relay Client

See also: The MercuryX Scheduling module and ETRN

MercuryC passes mail from your system to the outside world using SMTP (the Simple Mail Transport Protocol). MercuryC is a *relay client*: instead of delivering the mail directly to its final destination, it asks a nearby system to deliver the mail on its behalf. Relay delivery is particularly well-suited to dialup connections, and for use behind firewalls, since in each case, you are asking a machine which is better suited to the task to deliver the mail.

Configuring MercuryC

MercuryC is the module that the MercuryX Connection Scheduler can use to trigger remote SMTP queues via an SMTP extension called *ETRN*. Click here for more information on using ETRN.

Configuring the MercuryC SMTP Relay Client Module

See also: Logging and session logging
The MercuryX Scheduling module and ETRN

Smart host Enter here the name or address of the machine which MercuryC will ask to relay mail on its behalf. If you are using a dialup connection to an Internet Service Provider, then you will usually enter the address of one of your service provider's machines here - ask your Service Provider what to use. If you are behind a firewall, ask your network administrator which machine has SMTP access to the outside world through the firewall - that will be the machine to enter here. Any full SMTP server can be used for relaying, although it is common to apply controls to who can and cannot relay through systems.

Connect on TCP/IP port This is the port on the smart host to which MercuryC should connect. The standard port defined for this is 25, but in some cases (most notably if you are behind a firewall) you may have to enter a different port number here. Consult your ISP or Network administrator to find out if you need to alter the setting of this field.

Announce myself as The SMTP protocol has an identification phase, during which the client will tell the server its name. MercuryC will usually tell the server the name defined under *Internet name for this system* in the Mercury Core configuration dialog, but on rare cases you may need it to use a different name. If this is the case, enter that name here, otherwise leave this field blank. This option is inherently quite technical and you should enter a value here only if you are sure of what you are doing.

Delivery failure template The path to a template file which MercuryC should use to format delivery failures. The Mercury installer will create a default template file for you which should be suitable for most cases.

TCP/IP timeout the length of time in seconds that MercuryC should wait for data on a connection before assuming that the connection is no longer valid and aborting it. If you are connecting to a smart host that runs the Sendmail SMTP server, you may have to set this value to a quite large number (for instance, 600), because Sendmail insists on validating each address as MercuryC supplies it, which can take time.

Poll the queue every xx seconds How often MercuryC should check the queue to see whether there are any messages it should send.

Use extended SMTP features where possible SMTP is a very old specification, but in the last couple of years, a number of attempts have been made to modernize it. These modernization efforts are designed to be backwards-compatible - so, a client that uses them should still be able to communicate with older servers that do not understand the new features. In rare cases, you may encounter an old server that cannot cope with the new features; when this happens, you may have to uncheck this control. In general, this control should always be left checked unless you find you have specific reasons for unchecking it.

MercuryP, POP3 Server

MercuryP is a POP3 server - it allows users to retrieve their new mail without having physical access to the disk drive where it is stored .MercuryP presents a list of new mail messages to the POP3 client, which then decides which ones it wishes to download to the remote system.

If you are using Pegasus Mail as your electronic mail program, you will not usually use POP3 to retrieve your mail, because Pegasus Mail can read the mail directly from the directory on the file server where it is stored. There is nothing to preclude you from using MercuryP with Pegasus Mail, however, and if you are using other POP3 mail clients, they will interact with MercuryP.

Configuring MercuryP

Configuring the MercuryP POP3 Server module

See also: Logging and session logging

IP Interface to use If your computer supports multiple IP interfaces, you can use this field to tell MercuryP which interface it should select when listening for connections: enter the interface as a dotted IP address in the general form *www.xxx.yyy.zzz*. As an example, your system may have one IP address assigned to a dialup PPP connection, and another, different IP address assigned to a local Ethernet network - you would enter here the interface MercuryP should use. If you leave this field blank, MercuryP will listen on all available interfaces. Unless you are *very* sure of what you are doing, or have been instructed by an ISP or network administrator, you should leave this field blank. If you change the IP interface in this field, you must restart Mercury before the new interface number will be used.

Listen on TCP/IP port The TCP/IP port is the socket into which the POP3 client "plugs" to access MercuryP's services. The standard port for POP3 services is 110, but in rare cases (particularly if you use a proxy server) you may need to change this. Consult your ISP or network administrator to find out if you need to use an alternative POP3 port. If you are unsure, leave it set to 110.

Global profile settings

These settings control how Mercury should handle some aspects of connections from POP3 clients. These controls apply global settings - settings which will apply to all your users. You can also create user-by-user POP3 profiles to deal with the case where a specific user's POP3 client needs adjustment (see below for more information). In general, the default settings in this dialog will work with the majority of POP3 clients.

Mark retrieved mail as "read" When this control is checked, MercuryP will change the status of all messages downloaded by POP3 clients to "read". This can be used in conjunction with the "offer only unread mail" option to allow users to leave copies of all their mail on the server, but be offered only mail they have not seen.

Offer only unread mail If this control is checked, Mercury will list only messages whose status is "unread" - mail marked as read will be invisible to POP3 clients. This option is very convenient, but is also not a part of the POP3 standard.

Manufacture status headers Some mail programs (most notably Qualcomm's Eudora) rely on finding a non-standard mail header called "Status" in the headers of the message to determine whether or not a message has been read. The "status" header is generated by some unix mail servers, but is not a standard part of the Internet mail specification. MercuryP does not use status headers, but if this control is checked, it will create them "on-the-fly" for the benefit of POP3 clients that depend on them. There is usually no reason to uncheck this control, except that it adds a very small extra data overhead to POP3 downloads. If you are using Eudora as your mail client, this control should always be checked.

Ignore POP3 delete commands If you want to ensure that no mail is ever deleted from mailboxes, check this control. This tells MercuryP to disregard requests from clients to delete messages on the server. If this control is checked and Offer only unread mail is unchecked, then the user will be presented with the same mail messages every time he connects. Use this option with care.

POP3 deletions survive resets The POP3 standard states that when a POP3 connection terminates abnormally (for instance, because of nework problems), then the mailbox must be restored to the state it was in when the connection was established; all deletions must be undone and the status of all messages is returned to what it was at the time of connection. Checking this control tells Mercury that it should not delete messages that the POP3 client has marked for deletion even if the connection is closed abnormally. This option should be used with considerable care, but may be useful in situations where network stability is often unreliable.

Local profile settings

Local profile settings can be made on a per-user basis by creating a file called POP3.PRO in the user's new mail directory. POP3.PRO can contain any of the following statements:

```
Mark read
Show read
Show status
No delete
Delete is final
```

Each statement can be set to Y or N to enable or disable that setting. For example, to create a POP3 profile for a user that marks all downloaded mail as read and where deletions survive resets, you would add the following two lines to POP3.PRO:

```
mark read : Y
delete is final : Y
```

Statements missing from the file will use the default value determined by the Global profile setting controls (see above). Statements in POP3.PRO are not case sensitive.

MercuryD, Distributing POP3 Client

MercuryD is a POP3 Client Module designed to retrieve mail from as many remote hosts as you wish and to distribute that mail to users on your local system or network.

MercuryD can retrieve mail from a remote account and deliver it all to a single user, or, if the remote account is a so-called *Domain Mailbox*, where all mail addressed to any user at a specific domain is placed in a single mailbox, then MercuryD can distribute the mail from that mailbox to the appropriate local addressees by interrogating the address fields of each message.

You will typically use MercuryD instead of the MercuryS SMTP Server module, in situations where you want intermittent dialup access to the Internet (say, once every hour or so). The two are not incompatible, however, and there may be occasions where you might want to load both modules.

Configuring MercuryD

Configuring the MercuryD Distributing POP3 Client module

See also: Logging and session logging

Work directory Enter here a path to a directory where MercuryD can create temporary files during the download process. The directory should be on a volume with plenty of free space (at least 15MB is recommended).

Check every x seconds This setting controls the frequency with which MercuryD should go through the list of accounts checking them for new mail. For example, if you want MercuryD to check for new mail once per hour, you will enter 3600 in this field.

TCP/IP Timeout the length of time in seconds that MercuryS should wait for data on a connection before assuming that the connection is no longer valid and aborting it.

POP3 account information

This section contains the login information for each account MercuryD is to check for new mail. Each entry consists of a host, a username, a password, and the name of the local user who should receive the mail from the account.

Host The name or IP address of the machine to which MercuryD should connect via the POP3 protocol when checking the account for new mail.

Username The login name MercuryD should use when connecting to the POP3 server.

Password The password matching the username for the POP3 account

Local user If you enter the name of a local user on your system (one to which Mercury can delivery directly) then all the mail downloaded from the remote account will be sent to that local user, irrespective of the address fields in the message. If you leave this field blank, MercuryD will examine the *To*, *CC* and *BCC* fields of each message looking for addresses it recognizes as local. When it finds a local address, it will send a copy of the message to that local user. This facility allows you to have a single mailbox (called a *Domain Mailbox* by most Internet Service Providers) into which all mail for any users at a specified domain is placed; MercuryD can then retrieve the mail from that mailbox and route it to the appropriate local users for you.

Default user When distributing mail from a domain mailbox, MercuryD may encounter messages for whom it can find no local recipient; this will commonly happen if one of your users subscribes to a mailing list, since mailing lists usually do not indicate the actual recipient anywhere in the message headers. In cases such as this, MercuryD can be told to deliver the message to a specific, or *default* user. If you leave this field blank, MercuryD will discard any messages for which it can find no local delivery addresses. This field is only meaningful when you have told MercuryD to distribute mail (by leaving the Local user field blank).

To add an account to the MercuryD service list, simply click the *Add* button then fill in the six account controls and click *OK*. To edit an entry, click once on its entry in the list and click the *Change* button.

MercuryX Connection Scheduler

MercuryX can be used to schedule the operation of other protocol modules in the Mercury/32 system; it can also be told to execute certain commands before and after it activates the protocol modules under its command.

MercuryX is primarily intended for users operating in a dialup environment. Its operation is specially tailored to provide maximum flexibility with the shortest practical connection times.

Configuring MercuryX
Dialling considerations

Configuring the Mercury Connection Scheduler module

See also: Dialling considerations

Commands issued before and after connecting

MercuryX can be told to execute programs before it starts the Mercury Protocol modules and after it has shut them down during each rota period. You can enter any command in these fields, and the command can have a commandline. For maximum reliability, we recommend that you include a full path to the executable file in the commandline.

Run this command before starting If this field is not blank, MercuryX will attempt to run the command you specify before activating the Mercury protocol modules. Possible uses for this include invoking dialers, or loading network modules.

Wait until this process terminates before starting Mercury service processes If this control is checked, MercuryX will attempt to wait until the process in the *Run this command before starting* option has terminated before proceeding to activate the Mercury protocol modules. This option will work reliably with all Win32 applications, most Win16 applications, and some DOS applications. If this control is checked, MercuryX will not *wait X seconds* before starting the protocol modules - it will start them as soon as the process terminates.

Run this command after stopping If this field is not blank, Mercury will run this command after it has shut down all the protocol modules at the end of a rota cycle and waited the X seconds delay, if that is defined.

Before and after connections wait X seconds before running command If you enter a number in this field, MercuryX will wait that many seconds after invoking the startup command at the start of a rota cycle, and before invoking the shutdown command at the end of a rota cycle. If the *wait until this process terminates* control is checked, MercuryX will ignore this delay.

Use Win98/IE4 dialling functions Under Windows 98 and on systems where Internet Explorer v4.0 has been installed, Mercury can take advantage of dialling functions built-in to the operating system. If your system matches those described, you can tell Mercury to dial or hang up using these new functions by checking either of these controls. It is possible to "mix and match" options - so, you can use a command to dial, but the Win98 function to hang up if you have a need to do this.

Issue SMTP ETRN commands (RFC1985) to start remote queues Internet Standards Document RFC1985 defines a special command called *ETRN*, which can be issued by a dialup client to indicate that it is online and ready to receive mail. If your Internet Service Provider has a mail server that supports this command, then you can tell Mercury to issue it when it comes online - this is a useful way of scheduling when you will and will not receive mail from the Internet, but it requires the co-operation of your ISP. In order to use this option, you must also have either the MercuryC SMTP Client or the MercuryE SMTP Client Protocol Module installed in your copy of Mercury. For more information on ETRN and whether or not you can use it, please contact your Internet Service Provider (if they don't know what you mean when you mention "ETRN" or "RFC1985") then they probably don't support this option).

When using ETRN commands to start remote queues, you need to create a file called ETRN.DAT in the directory where Mercury is installed. Clicking the *Specify* button in this dialog will create this file if it does not exist, or will edit your existing ETRN.DAT. The format of the file is quite simple and is documented heavily in comments when it is created.

Allow queues to "drain" completely before shutting down connection This setting only affects the client modules, MercuryC (or MercuryE if you have installed that option instead) and MercuryD. When it is checked, once MercuryX reaches the end of a connection cycle, it will wait until the client processes enter

an idle state by themselves before proceeding to shut down the connection. This means that you can tell MercuryX to use a one minute cycle once per hour, but all mail in the queue will still be sent even if it takes fifteen or twenty minutes: as soon as all mail in the queue has been processed, MercuryX will close down the connection. If this control is not checked, then MercuryX will ask the client modules to close down after the job they are currently processing is complete: in this case, mail can be left in the queue until the next cycle for processing.

Process control mode This setting determines how MercuryX should handle busy processes when it comes time to terminate a scheduled connection cycle. The setting that you should use depends very much on the way you use Mercury - essentially, it allows you to control how the Mercury client modules (MercuryC, MercuryE and MercuryD) are instructed to go offline, and also tells MercuryX how to handle the Mercury Server modules (MercuryS, MercuryP and MercuryH).

No control When this option is selected, all Mercury modules are instructed to go offline at the end of the connection cycle. Servers will go offline at once, and client modules will complete the job they are currently processing. Jobs that the client modules have not yet processed will remain in the queue until the next scheduled connection cycle.

Clients When this option is selected, MercuryX will wait until all clients indicate that they are idle (i.e, have no further jobs to process) and will then take them offline. The connection will not be terminated until all client modules indicate that they are idle and have been shut down. In this mode, Server modules are **not** instructed to go offline, but will continue listening for connections. This mode is useful if you use the client modules to connect to the outside world and the server modules to handle requests on your local area network. This mode is particularly suited to environments where dial-on-demand routing or ISDN is used.

Clients/servers When this option is selected, MercuryX will wait until all clients indicate that they are idle before shutting them down. It will also wait until all server modules are idle before terminating the connection, but will not actually instruct the server modules to go offline. This mode is intended to handle cases where the server modules may be able to accept connections from both the outside world while the connection is established, and from your local area network at other times.

Scheduling details

MercuryX allows you to create different schedules for each day of the week. Each day can have a *peak time* and an *off peak time* - the assumption is that peak time connections will be more frequent and will last longer than off-peak connections. Defining a scheduling *rota*, or a set of times for a given day, is simplicity itself - simply select the day from the drop-down control, then indicate the peak times in the *Between XXXX and XXXX* fields; once you have done this, indicate how often MercuryX should start the protocol modules and for how long, then do the same in the remaining fields for the off-peak times. Note that the connection cycle includes the activation time - so, if you tell MercuryX to start processing every five minutes for two minutes, it will begin a new connection three minutes after it shuts down the current cycle.

To copy the definition from another day into the current day, click the *Copy from* button and mark the day from which you wish to copy settings.

Dialling considerations

The process of dialling and hanging up intermittent Internet connections is one of the most frustrating and complex issues in the Windows environment.

Properly speaking, dialling and hanging up are functions of the Windows networking component that provides TCP/IP protocol support. This module, called WSOCK32.DLL, is a Microsoft-supplied component that is a built-in part of Windows. Unfortunately, it does not work correctly, and is unlikely ever to do so - Microsoft have shown no inclination to address its quite significant shortcomings. To explain why dialling and hanging up are system functions and not application functions, consider the situation where Mercury/32 is running at the same time as the user on the workstation is accessing the Internet using a web browser. If Mercury hangs up the connection, then the web browser will also be disconnected; similarly, if the user closes down the web browser and it hangs up the connection, Mercury will be cut off in mid-stream. Clearly, the system-level Network module, WSOCK32.DLL (which is used by both Mercury and the browser), is the only component in the system that knows how many tasks are active and hence when it is appropriate to close the connection.

At the time of writing, the Microsoft WSOCK32.DLL supplied with Windows 95, 98 and NT can initiate a dialup connection correctly, but will not correctly hang it up when it is idle.

Microsoft's failure to make WSOCK32.DLL handle dialling and hanging up correctly has meant that application developers have had to come up with their own solutions to the problem. In general, these solutions take two forms: writing calls to the Windows RAS subsystem to force dialling and hanging up, and using functions in a special Microsoft Internet Explorer 4.x module called WININET.DLL to force dialling and hanging up. Mercury/32 supports both these approaches.

1: Making RAS calls Under Windows NT, you can use the MercuryX scheduler module's command options to use the Windows NT *RASDIAL* utility to establish and disestablish connections. Alternatively, you can use a free version of RASDIAL written by Claudio Fahey, called RASDIAL95. This utility, which works under Windows 95, 98 and NT, is included with Mercury in the EXTRAS subdirectory of the directory where you installed Mercury/32. The utility is easy to use and has a comprehensive readme file describing its operation. We wish to offer our appreciation and thanks to Claudio for allowing us to include RASDIAL95 with Mercury/32.

2: Using WININET calls If you have installed Internet Explorer 4.0 on your system, or you are using Windows 98, then MercuryX can take advantage of special functions provided on these systems to establish and disestablish Internet connections. To enable this option, check one or both of the controls associated with it in the MercuryX configuration dialog.

MercuryH PH Directory Server

MercuryH is a directory services module - it accepts requests for addressbook lookups from other systems and returns lists of possible matches. The protocol used by MercuryH is called the *PH protocol*, also sometimes known as the *QI/CSO protocol*: it is widely-used on the Internet by mail clients such as Pegasus Mail and Qualcomm's Eudora. At the time this help file was written, an Internet standard was in the late stages of being formalised for the PH protocol - MercuryH implements that standard.

In order to fulfil queries, MercuryH uses a Pegasus Mail addressbook file, mapping the fields in the addressbook file onto the standard fields defined by the PH protocol. Any Pegasus Mail addressbook can be used by MercuryH, and you can use the import/export tools provided with Pegasus Mail to create and maintain addressbooks for MercuryH.

Configuring MercuryH

Configuring the Mercury PH Query Server

See also: Logging and session logging

Addressbook file: enter here the path to the Pegasus Mail addressbook file MercuryH should use when resolving queries. Pegasus Mail addressbooks consists of two files with the same name, one with the extension .PMR, the other with the extension .PM!. MercuryH only needs access to the .PMR file - enter the path to this file in this field.

MOTD file: The PH protocol allows you to define an arbitrary text message (referred to as a *Message Of The Day*, or *MOTD* file) that is sent in response to a PH *status* command. Enter the name of the text file MercuryH should send when it receives a status command here. The file should be plain text, with lines 60-70 characters in maximum length. This field is optional -you do not have to provide a MOTD file. You can perform simple text editing on your MOTD file by clicking the *Edit* button after you have entered the path.

Admin address: The PH *status* and *siteinfo* commands can advertise the address of an administrator to whom requests for support should be sent. If you wish to have a PH server administrator, enter his or her full e-mail address in this field. As with the *MOTD file* field, this field is optional.

TCP/IP Timeout the length of time in seconds that MercuryH should wait for data on a connection before assuming that the connection is no longer valid and aborting it.

IP Interface to use If your computer supports multiple IP interfaces, you can use this field to tell MercuryH which interface it should select when listening for connections: enter the interface as a dotted IP address in the general form *www.xxx.yyy.zzz*. As an example, your system may have one IP address assigned to a dialup PPP connection, and another, different IP address assigned to a local Ethernet network - you would enter here the interface MercuryH should use. If you leave this field blank, MercuryH will listen on all available interfaces. Unless you are *very* sure of what you are doing, or have been instructed by an ISP or network administrator, you should leave this field blank. If you change the IP interface in this field, you must restart Mercury before the new interface number will be used.

Listen on TCP/IP port By default, MercuryH listens for connections from the outside world on port 105, which is the standard reserved port for the PH Query protocol. In some cases, particularly when you are behind a firewall, you may wish to listen on an alternative port - enter the number of that port in this field. **If you change this field and save the dialog, you will need to exit and restart Mercury/32 before the change will take effect**.

Connection control

The *Connection Control* section allows you to place restrictions on the hosts from which MercuryH will accept connections. To add an entry to the list, type its IP address in the *IP Address* field and select either *Allow* or *Refuse*, then click *Add*. The digit 0 acts as a wildcard in a connection control entry, so adding an entry refusing access to 165.25.9.0 will cause MercuryH to refuse connections from any machine whose address's first three octets are 165.25.9. Note that there is an implicit rule "Allow 0.0.0.0" at the end of this list, so if an address "drops through" the list, it will be automatically accepted.

To edit a connection control entry, highlight it in the list, edit the *IP Address* and *Refuse/Allow* controls, then click the *Change* button.

Support for Local Area Networks

Mercury supports Local Area Network architectures such as Novell NetWare via a system of plugin modules that give it access to the services offered by the Network. Using a Network plugin allows Mercury to take advantage of preconfigured mailboxes, built-in user databases (such as Novell NDS, or the NetWare Bindery) and generally reduces the administration load associated with maintaining mail users.

Mercury currently has separate plugin modules for Novell NetWare 3.x servers, using the Novell NetWare Bindery as a user database, and for Novell NetWare 4.x servers using Novell's NDS user database.

Novell NetWare 3.x (Bindery mode) support
Novell NetWare 4.x and 5.x (NDS mode) support

A network plugin will be supplied as a Windows DLL with the name "MN_xxx.DLL", where "xxx" is a three-character code describing the network supported by the module. For example, the DLL providing support for Novell NetWare 3.x is called MN_NW3.DLL. Exactly one Network support plugin can be installed at any time - Mercury/32 will use the first plugin it finds in the same directory as MERCURY.EXE.

When a Network plugin is installed, it may make configuration options available via the *Network support* option on the Mercury *Configuration* menu.

Novell NetWare 3.x (Bindery mode) support

Using this plugin (MN_NW3.DLL), Mercury can act as a mail server for one or many Novell Netware 3.11 or later file servers, using the file server's Bindery as a user database. In NetWare mode, Mercury is designed to interoperate closely with Pegasus Mail, a mail client by the same developer. It will deliver mail to the new mail location expected by Pegasus Mail, and Pegasus Mail can be configured to send mail to the Internet by placing it directly into Mercury's mail queue.

Under NetWare 3.x, all users are automatically given a mailbox directory in the file server's SYS:MAIL directory. Mercury is able to find and use this mailbox for delivering mail. When you create a user using the NetWare SYSCON command or by any similar means, the user's mailbox is automatically created and can be used immediately without any further configuration.

In NetWare mode, Mercury can use certain feature extensions, such as mail forwarding, which are manipulated using the Pegasus Mail PMGRANT program - see the Pegasus Mail documentation for more information on setting and using these features.

By default, in NetWare mode each user's address will be their NetWare login name @ the Internet name of the system. You can provide users with custom addresses that differ from this scheme by creating synonyms for them.

Multiple servers

Mercury can act as a mail transport for multiple Novell NetWare 3.x servers. Each file server must have an unambiguous domain name - so, you cannot have two servers both using the domain name "myhost.mydomain.com"; each server must have its own Internet name, unless you are willing to create aliases for every user who is to receive mail. When serving multiple servers, Mercury needs access to a privileged account on each server. You tell Mercury about the account and password it should use for each server using the *Network Support* option on the Mercury *Configuration* menu. The information is stored in an encrypted file called NETWARE3.DAT in the same directory as MERCURY.EXE.

If you wish to create a "group domain" (i.e, a situation where two separate file servers have the same Internet domain name) then you must create aliases for every user who is to receive mail. The alias should contain the user's e-mail address as the alias, and the real address must in the form SERVER/USER (the same as you would use to login to the server as that user). You must still have an Mercury access entry defined for every server using the *Network support* configuration option.

Bindery mode on NetWare 4.x and 5.x servers

Novell NetWare 4.x and 5.x file servers have support for a subset of NetWare 3.x's bindery, and the Mercury Bindery mode plugin can be used with those servers (although we would normally recommend that you consider using the NDS mode module). You should be aware, though, that NetWare 4.x and 5.x do not create the user's Bindery mode mailbox until the user logs into the server in bindery mode, using the command login /b, or by specifying a bindery login in the NetWare Windows client. It is important that your users login in Bindery mode before using the mail system if you run in Bindery mode on a NetWare 4.x or 5.x server, or otherwise mail delivery may fail.

Novell NetWare 4.x and 5.x (NDS mode) support

With the introduction of NDS under NetWare 4.x, Novell discarded the simplicity of NetWare 3.x and replaced it with a much more complex and administration-intensive system. They also removed most of the features of NetWare 3.x that allowed Pegasus Mail and Mercury to operate in an administration-free manner under that system.

Configuring Mercury and Pegasus Mail to work under NetWare NDS is slightly more complex than in Bindery mode. You need to create mailboxes for each user, and administration of issues such as synonyms and user features is also rather more complex, owing to the nature of NDS itself.

Mercury/32 is supplied with a NetWare NDS mode plugin called MN_NW4.DLL, suitable for use with both NetWare 4.x and 5.x file servers. This plugin is intended to complement the operation of Pegasus Mail in NDS mode, and assumes that you have already downloaded and installed the Pegasus Mail NDS enabler. This file is available from any of the standard Pegasus Mail download sites: at the time of writing, the current version of the NDS enabler was v2.1, available as WPMNDS21.ZIP. This file contains a comprehensive guide to NDS mode operation called README.NDS, and a powerful NDS-mode configuration and maintenance utility called NCONFIG.EXE. We recommend that you download this file and familiarize yourself with the documentation and with NCONFIG before installing Mercury in NDS mode.

In NDS mode, Mercury takes advantage of NDS to provide an extremely flexible solution, at the cost of some extra administration and maintenance.

For more information on this, and for downloading the enabler, please see our web site, http://www.pmail.com, or http://www.pmail.gen.nz.

Address Synonyms

Many sites need to use custom address formats where the user's e-mail address is quite different from his actual user name on the system. For example, you may want your addresses to have the form *Firstname.Lastname@host.domain*. Simple aliasing won't allow you to do this kind of addressing easily - it will handle incoming mail well enough, but in order to get mail going out from your system to use the alternative address forms requires the co-operation of your mail client.

Pegasus Mail and Mercury/32 can combine to support alternative address forms like this (referred to as *synonyms*) using a special database called a synonym database: Mercury needs this database so it can work out the recipient for incoming mail, while Pegasus Mail needs it to work out what address it should write into outgoing messages. Using synonyms differs slightly depending on whether or not you are running in Novell NetWare mode:

** If both Pegasus Mail and Mercury are running in NetWare mode:* create your synonyms using the Pegasus Mail PMGRANT (or NPMGRANT in NDS mode) commandline program, then create the Mercury synonym database using the CH_SYN.EXE (or NSYNONYM.EXE in NDS mode) commandline program. Copy the synonym database somewhere accessible by Mercury, and make sure the *Synonym database file* entry of the *Files/Directories* preferences page of the *Mercury core module* configuration dialog refers to that file. Pegasus Mail will automatically pick up each user's synonym from the NetWare user database.

Tip: The Pegasus Mail NDS mode configuration utility, NCONFIG.EXE, can greatly simplify the creation and maintenance of synonyms in NDS mode, both for Pegasus Mail and for Mercury.

** If either Pegasus Mail or Mercury is running in non-NetWare mode*, then you will need to follow these steps:

1: Create a *synonym source file*: this is a text file where each line has the form

```
synonym == username
```

So, on the left-hand side you place the user's e-mail address as you want it to appear in outgoing mail, and on the right hand side his or her local user name. Note that each line must begin hard against the left-hand margin.

2: Compile the source file using the FSYNONYM commandline program. FSYNONYM takes your input file and creates a synonym database file.

3: Copy the synonym database somewhere accessible by Mercury, and make sure the *Synonym database file* entry of the *Files/Directories* preferences page of the *Mercury core module* configuration dialog refers to that file.

4: Copy the synonym database file into the same directory as your copy of the Pegasus Mail executable file (either WINPMAIL.EXE or WINPM-32.EXE), making sure it is called SYNONYM.MER.

5: [Only required once] Use the Pegasus Mail option on the Mercury configuration menu to configure your copy of Pegasus Mail. This operation is only required if you have upgraded your copy of Mercury from an earlier version, and only needs to be done the one time: it creates a new version of PMGATE.SYS that instructs Pegasus Mail to use the synonym file if it exists.

Note: in order to use synonyms in non-NetWare mode, you must be using Pegasus Mail for Windows v3.01b or later. Earlier versions will not recognize the synonym database and will use the user's regular e-mail address instead.

Daemons (third-party extensions)

One of Mercury's most powerful and least visible features is its *Daemon interface* (*Daemon* is a term borrowed from the unix world meaning a *resident process*). This interface allows third-party developers to create extensions to Mercury for processing mail. Using the interface, a Daemon can accept mail for any or all addresses on the server, can send and parse complex MIME messages, and can generally perform nearly any imaginable task that you could do with e-mail.

The Daemon interface is extremely rich, but is not particularly difficult to program if you have a little experience writing Windows code. If you are interested in reading more about the Daemon interface, please see the files DAEMON.TXT and DAEMON.H in the RESOURCE subdirectory of the directory where you installed Mercury/32.

Logging and session logging

Most Mercury protocol modules can keep logs of the work they do. Logging comes in two forms, *General Logging*, which is simply a short record of each transaction maintained by the server modules with one transaction on each line in a file, and *Session Logging*, which is a byte-for-byte transcript of the entire exchange between a server and a client.

General logging is supported by MercuryS (the SMTP server), MercuryP (the POP3 server) and MercuryH (the PH Directory Services server). Each of these modules can be given a filename for a general log file: they will create this file as required and will write a single-line record describing each connection they process. Each module's general log file has a regular format that can be parsed by software tools to generate statistics and reporting. To turn general logging off, enter a blank filename. You can change the name of the general log file any time you wish in order to start a new log file.

Session logging is supported by MercuryS, MercuryP, MercuryH, MercuryC (the SMTP client module) and MercuryD (the POP3 client module). When session logging is turned on in a module, it will create one file per connection in a directory you specify. The file will contain a complete transcript of the data sent between the module and the remote machine to which it is connected. Session logging files are given numerically ascending filenames in the general form `TCPxxxx.yy`, where `xxxx` is a serial number and `yy` is an extension representing the module that created the log file. The extensions generated by the various modules are as follows:

`TCP*.MS`	Generated by the SMTP server, MercuryS
`TCP*.MC`	Generated by the SMTP client, MercuryC
`TCP*.MH`	Generated by the PH server, MercuryH
`TCP*.MP`	Generated by the POP3 server, MercuryP
`TCP*.MD`	Generated by the POP3 client, MercuryD

You can turn session logging on and off at any time using the *Enable session logging* checkbox in the relevant configuration dialog for the module. The setting of this checkbox is remembered by each module across sessions.

NOTES You should be aware of some important issues with session logging:

1: The *Session logging* parameter in each module's configuration dialog refers to a subdirectory, not to a filename. If the subdirectory does not exist, Mercury will create it as required. You can have each module create its session logs in a different subdirectory, or, if you prefer, you can have them all write their logs to the same directory.

2: The session log file includes all the data sent across the connection. So, the session log file for the arrival of an incoming mail message that is 5MB in size will be at least 5MB. Keep this in mind when you allow for disk space on your system - session log files can be large and can consume disk space at a startling rate.

3: Session logging slows connections down - not by much, but it does have an impact on the rate at which mail is processed.

4: When session logging is turned on, all data sent across the link is included: in the case of the MercuryD POP3 client and the MercuryP POP3 server, this means that passwords will be stored in the session log files as plain text. This may create a security risk and should be carefully considered.

Internet name for this system

Enter here the Internet domain name for the machine on which Mercury is running. Mercury will use this information when forming certain addresses, such as the postmaster address. The name you enter here should be a fully-qualified domain name; if you are intending to use Mercury to provide mail services outside your immediate organization, the name you provide will need to be accessible in your Domain Name Server (DNS) system.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Mail Queue Directory

Mail queue directory, SMTP queue directory These entries control where Mercury should look for and place mail that is to be processed. The mail queue is where mail clients such as Pegasus Mail put messages for Mercury to process; Mercury also places jobs here on occasions, usually when generating autoreplies and mailing list mail. The SMTP queue is the location where the Mercury Core Module should place messages intended to be sent to the outside world by the MercuryC Client module. The mail queue and SMTP queues can be, and normally are the same.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

SMTP Queue Directory

Mail queue directory, SMTP queue directory These entries control where Mercury should look for and place mail that is to be processed. The mail queue is where mail clients such as Pegasus Mail put messages for Mercury to process; Mercury also places jobs here on occasions, usually when generating autoreplies and mailing list mail. The SMTP queue is the location where the Mercury Core Module should place messages intended to be sent to the outside world by the MercuryC Client module. The mail queue and SMTP queues can be, and normally are the same.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Local Mailbox Directory Path

This entry is ignored if you are using a network support module: it tells Mercury how to locate your users' mailbox directories. The string is a standard pathname containing one of two special placeholders - either *~8* or *~N*. When Mercury uses the string to find the mailbox for a user, it replaces *~8* with the first eight characters of the user's name, or replaces *~N* with the user's whole username. If you are using Pegasus Mail v3.01d or earlier, or any 16-bit Pegasus Mail client as your mail client in conjunction with Mercury, you should not use the *~N* substitution - you should only use the *~8* version. If this 8-character restriction creates problems with usernames for you, you could consider defining synonyms for the names that are longer than 8 characters, or upgrading to a later version of Pegasus Mail.

NOTE: It is currently a restriction of Mercury that the ~8 or ~N placeholder must appear at the end of the path - so, *C:\PMAIL\~8* is legal, but *C:\PMAIL\~8\MAILBOX* is not.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Time Zone, Auto timezone

Time zone Enter here the timezone for your site, expressed as a plus or minus difference from GMT. So, if you are in Los Angeles and are currently at GMT - 9 hours, you would enter

-0900

in this field. Mercury will accept the so-called "vernacular" time zone format, such as PST and CST, but the use of these formats is no longer recommended on the Internet and we strongly advise you to avoid them, since their use makes it impossible for most mail programs to sort properly by date.

If the *Auto* control is checked, Mercury will work out the necessary timezone correction automatically using information supplied by the Operating System, and anything you enter in the *Timezone* field will be ignored.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Poll for new mail every x seconds

This setting controls how often the core module should check to see if there is mail waiting to be processed in the queue. We recommend that you do not set it below ten seconds for performance reasons.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Username of postmaster

Every system capable of receiving Internet mail must have a user called *postmaster*, to whom problem and status reports are sent. The postmaster account is usually an alias to a real user on your system, and this is the expectation within Mercury. Enter in this field the username of the user on the machine where Mercury is running who is to act as your postmaster. While it is permissible to have a non-local address as your postmaster address, we ***strongly*** recommend that you do not do this, since it can create real problems and mail loops when the remote machine is unreachable. This setting is mandatory - Mercury cannot run properly without it.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

For delivery failures return x lines of the message

When Mercury cannot deliver a message to a local user for whatever reason, it will invoke a template file you provide for delivery failures. One of the optional replacements that can be used in the delivery failure template file is a special substitution that sends a certain number of lines from the failed message. This configuration option controls how many lines of the message are returned when the special partial return substitution is encountered.

See also:

- The Mercury Core Module configuration
- Configuring Template Files

Press <Esc> to return to Mercury.

Broadcast notifications for normal mail

Mercury has special Network awareness modules that allow it to take advantage of certain specific features of some local area networks. One of the features that some networks (such as Novell NetWare) support is the transmission of a single-line broadcast message that appears on the target user's screen. If this control is checked and you are running Mercury on a network that supports broadcast messages, Mercury will send a short message to users when new mail arrives for them.

See also:
The Mercury Core Module configuration
Network Support in Mercury

Press <Esc> to return to Mercury.

Broadcast notifications for receipts

Mercury has special Network awareness modules that allow it to take advantage of certain specific features of some local area networks. One of the features that some networks (such as Novell NetWare) support is the transmission of a single-line broadcast message that appears on the target user's screen. If this control is checked and you are running Mercury on a network that supports broadcast messages, Mercury will send a broadcast message advising the arrival of mail messages that confirm reading or delivery.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Send copies of all errors to the postmaster

If this control is checked, Mercury will send a copy of all error reports it generates to the local postmaster as well as to the original sender of the message. This allows the postmaster the option of correcting addressing errors and other simple problems.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Change file ownership to recipient

As with broadcast notifications, some Network systems support the idea of file ownership, usually to calculate disk space usage. If your network supports this idea and this control is checked, then Mercury will attempt to change the ownership of all the messages it delivers so that the actual recipient owns the file.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Suppress validation of From field when processing mail

Mercury usually attempts to validate that the "From" field of all mail it delivers locally is legal. This can sometimes cause problems if you receive mail from sites that use broken or faulty mail programs; if this is the case, you can suppress the validity check Mercury performs by checking this control.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Hard to quit (exit only on Ctrl+File|Exit)

When this option is checked, Mercury will ignore all attempts to quit from it, and will minimize itself to the system tray instead. In order to quit from the program, choose "Exit" from the "File" menu while holding down the Ctrl key. This option is useful when Mercury is run on a server to prevent people from accidentally closing it down.

See also: The Mercury Core Module configuration

Press <Esc> to return to Mercury.

Mail filtering rules

Like its companion product, Pegasus Mail, Mercury allows you to perform automated processing of mail when messages matching particular conditions are met. You might use this, for example, to forward messages, or print them automatically for certain recipients, or to delete unwanted "spam". The Mercury core module is responsible for applying filtering rules, which it does as the first stage in delivering a message.

Using global rules
Creating and using general rules

You can trigger your rules (that is, define the set of conditions which must be true before the rule will be applied to a message) based on a number of criteria. You can select a number of types of rule trigger using the buttons at the left of the rule editor dialog; the possible trigger types include the following:

Matching on text in standard message headers
Matching on regular expressions
Matching based on list membership

There are also buttons that create rules that always trigger, rules that trigger on the size the message, rules that trigger depending on certain predetermined characteristics of the message, rules that are simply comments and have no effect on rule processing, and rules that are labels (see *Advanced formatting*, below, for more information on why you might want to use a label). Finally, you can use the *Not* button to change any rule so that it only triggers if the condition it describes is not met in the message.

A wide range of actions can be triggered by a rule. Rule processing continues until all rules have been applied, or until the message is moved to another user or deleted as the result of a rule.You can define multiple rules with the same trigger conditions to have multiple actions applied to the same message -- Mercury will apply them in the order in which they appear in the list. Make sure that any rules containing "Move" or "Delete" actions are the last you define for a particular trigger text, since rule processing on a message stops after these actions.

Actions that can be triggered by a rule

Advanced filtering - flow control and logical operations

Global rules

Global rules are a single set of rules applied to all your incoming and outgoing mail. You can create and maintain your global rule set using the *Edit global rules* option on the *Filtering rules* submenu of the *Configuration* menu.

Global rules allow you to create custom actions for all your outgoing mail - for instance, if you want to keep an audit trail of all the mail coming into and leaving your site, you could do this with a single global filtering rule.

The more global filtering rules you define, the longer it will take Mercury to process every message, even messages that do not trigger any rules. This usually isn't a problem, since electronic mail seldom needs to be immediate, but it is something to bear in mind when you create your rule sets.

If you find yourself needing to perform complex tasks that require large numbers of rules, you could possibly consider arranging to have a Mercury Daemon developed for you. Daemons are resident processes that can bring the full power of your PC to bear on specialized mail processing requirements: because they are compiled programs, they are inherenly very fast, but they do require some Windows programming.

Back to the main Mail Filtering help screen

Creating and using general rule sets

General rule sets are sets of filtering rules that can be attached to any e-mail address in your system via an alias. Unlike global rules, which are applied to all mail messages passing through the system, general rules are only applied when mail is delivered to the address to which they are attached.

You can create or maintain a general rule set using the options on the *Filtering rules* submenu of the *Configuration* menu. We recommend you use the extension .RUL to identify your rule sets, but this is not a hard-and-fast requirement of the program.

To use a general rule set you have created, you need to create an alias that tells Mercury to invoke the set. To do this, choose *Aliases* from the *Configuration* menu and create a new alias. For the *Alias*, type in whatever address is to be associated with the set, then for the *Real address*, type in the special string `FILTER:` followed by the filename of the filtering rule set.

Example: if you wanted to invoke the alias set `C:\MERCURY\ORDER.RUL` any time a mail message was sent to `orders@sales.com`, then you would create the following alias:

Alias	**`orders@sales.com`**
Real address:	**`filter:c:\mercury\order.rul`**

Using aliases to trigger your filtering rules ensures the security of the system, since only you can create an alias, and thus only you can choose which rule set is associated with that alias.

Filtering and real addresses Filtering is done before user delivery occurs, so if you wish, you can create a filtering alias that is the same as a real e-mail address on your system, and the filtering will occur instead of delivery. If you do this and still want the message delivered, you must make sure you place a *Copy to local user* rule that always triggers in your rule set, because Mercury does not otherwise deliver a message once it has been processed by the filtering rule subsystem.

Back to the main Mail Filtering help screen

Matching on text in predefined message headers

In the majority of cases, you will simply want to detect messages which contain particular addresses, or which have particular text in the subject field. Mercury provides an easy way of triggering rules based on conditions like this.

Click the Headers button at the left of the rule editor dialog, then simply check the fields in which you would like the test to be made and enter the text you would like to match in the "Contains this text" field. You can check as many of the six controls as you wish, although some are probably mutually exclusive (such as "From:" and "Subject:"). You should usually check at least one control, although it is permissible to check none (this is a useful way of disabling a rule without deleting it).

Exact matching: Mercury usually triggers the rule if any of the fields you check contains the trigger text anywhere; if you want Mercury to trigger only on an exact match, check the control labelled *Exact match*. Doing this means that the rule will only trigger if the header and the match are the same length and contain the same characters. The match is always case-insensitive -- this cannot be changed even for an exact match.

Examples:

If "Subject:" is checked and the trigger text is "SUBSCRIBE" then the rule will trigger if the subject is any of the following:

Subscribe
I want to subscribe to your list
Notice to all subscribers

If "Subject:" is checked and *Message field must match this text exactly* is checked, then the rule will trigger only for the first of the examples shown above ("Subscribe")

Using regular expressions as rule triggers

Using regular expressions to trigger rules is more complicated than matching on predefined headers but is considerably more powerful, since it allows you to match on any message header, or even on the contents of the message body.

Clicking the *Expression* button in the rule editor creates a rule that searches using a regular expression. You can restrict the extent of the search using the scope controls - ***NOTE*** you should be careful when using either of the scopes which permit searching the message body, since this can dramatically increase the time it takes Mercury to process the message (sometimes by a factor of as much as 10 times or more). You need only have one action whose scope is the message body in your entire rule set to cause this increase in processing time, although subsequent rules in the message body scope do not introduce appreciable further delay.

Next, enter the regular expression you wish to use in the "Regular expression" field. Mercury recognizes the following metacharacters in expressions:

*	Match any number of any characters
?	Match any single character
+	Match one or more occurrence of the last character
[ ]	Encloses a group of characters to match. Ranges can be specified in the group using '-'.

All metacharacters can be used as many times as necessary. Regular expression searches are always case-insensitive. To search for literal occurrences of any of the characters *, ?, + or [, you must enclose them in group markers (so to search for a literal asterisk, enter [*]). Regular expressions do not cross line boundaries - you can only perform expression matching within individual lines in the message. If the first character of your expression is not '*', then matching must begin at the start of the line - so if you want to use a regular expression to find lines containing text at any position, you must use a leading '*' character (this is the reverse of matching on predefined headers, where the match is always open unless exact matching is specified).

Note: Unlike predefined header text matching, regular expression matches do not match substrings - they match entire lines. Because of this, if you want to detect a sequence of characters occurring anywhere in a line of text, you must surround the text with a pair of asterisks (*). This is the regular expression way of saying that any text can precede and follow the search text. *Example:* if you want to match on any line containing the string *offer* using an expression, you would need to enter the match text **offer**. Similarly, if you wanted to match any line starting with the word *Subject* and containing the word *offer* anywhere else on the line, you would need to enter the match text *Subject*offer**. If you omitted the second asterisk, you would be telling Mercury that the line would have to *end* with the word *offer*.

Examples:

MIME-VERSION:*[12]*
or MIME-Version: +[12]*
Matches "MIME-Version: 1.0"
but not "MIME-Version: 3.0"
(The '+' matches multiple successive spaces)

pmail.gen.nz
Matches any line containing "pmail.gen.nz"

Priority:*urgent
Triggers a rule for urgent mail

Filtering messages based on list membership

With this type of filtering rule, the rule will trigger if the sender's e-mail address can be located in a Mercury distribution list. When you create the rule, simply type in the name of the distribution list Mercury should search, and it will do the rest (note: you should enter the name of the list as it appears in your "list of lists" file, without a domain).

You can tell Mercury to scan a plain text file instead of a Mercury mailing list by entering the special character '@' followed by the full path to the file. This approach can be a useful way of maintaining traffic lists or kill files.

This type of rule has two major applications:

1: Creating "kill" files to catch "spam" (Unsolicited Commercial E-mail) from known addresses. When you receive an unsolicited spam message, you can add the sender's address to a list of known evildoers, then delete all future messages from that address using a single rule of this type.

2: Verifying that a person is a list member: if you offer services that are triggered by filtering rules (for instance, if you return product information or encryption keys in response to automated messages), then you may wish to verify that the person sending the request is actually a member of a list of authorised people before providing the service. You can use a rule of this type to determine whether or not the person is authorised based on their membership of a list.

Advanced option

You can create an entry in your target distribution list that matches any address from a single domain by editing the list manually and adding a line exactly like this:

\MATCH *@domain.com

into the list. The "\" character must appear hard against the left margin of the file.

Example: to suppress all mail from any address within the domain "`bigdeals.com`", you would add the following line to your distribution list:

```
\MATCH *@bigdeals.com
```

Actions that can be triggered by rules

When a rule's trigger text matches text in the message, its action is performed. Actions are selected from the list in the Rule Editor - selecting an action may result in you being prompted for more information, such as selecting a destination folder, or entering a file name; you can always change the parameters for a rule at any time by clicking the *Set* button.

Rules are processed until a rule is triggered whose action results in the message being either deleted or moved to another user. Because of this, you should always take care to ensure that rules containing Move or Delete actions are the last ones you define for a particular trigger text.

You can also force rule processing to stop for a specific message by using the *Exit this rule set* action.

Most of the rule actions are pretty obvious, but the first two, *Copy to another user* and *Move to another user* require a little explanation. These rules make exact duplicates of the message in the mailbox of another local user - they bypass any autoforwarding or autoreply mechanisms in place, and unlike forwarding, do not change the message in any way at all. These actions are primarily intended for sites that need audit trails of the messages they send and receive. Note that you can enter a directory into which the messages should be placed instead of a user by entering the special character '@' followed by the full path to the directory (which must exist). Mercury ensures that no filename collisions occur in the directory where the messages are placed in any event.

The *Logical AND operator* rule action can be used to create groups of rules where all the rules must be triggered before the final action is applied; this process is called *logical operation* - click here for more information on logical operations.

Advanced filtering - flow control and logical operations

This section covers specialised uses of mail filtering and is intended mainly for advanced users. Please ensure you have read all the other sections on mail filtering before attempting to use the information in this topic.

Flow control

Many times, you may find that there are certain groups of rules that you want to apply repeatedly in a rule set, or that you want to have more control over the order in which rules are processed. This concept is called *flow control*, and Mercury provides six rule actions to support it - skip, exit, labels, call/return and goto.

Using flow control

Logical operations

Often, you may only want to apply a rule to a message if all of a number of conditions are matched, or if any one or more of a number of conditions apply. This kind of operation is known as a *logical operation* (it is also known by the technical name *Boolean operation*). Mercury implements logical operations by a combination of rule order and flow control structures. Before reading about logical operations, we strongly suggest you read the section above on flow control.

Creating logical operations

Using flow control in your rule sets

Many times, you may find that there are certain groups of rules that you want to apply repeatedly in a rule set, or that you want to have more control over the order in which rules are processed. This concept is called *flow control*, and Mercury provides six rule actions to support it - skip, exit, labels, call/return and goto.

Skip The simplest flow control operator is the *Skip next rule* action: when a rule triggers and this action is indicated, Mercury will skip over the next rule in the list without testing or applying it. You can use this as a way of handling exceptions to a general rule - for instance, imagine that you want to delete all messages where the subject contains the phrase *free offer*, except when that message comes from the address *support@pmail.gen.nz* - you would add the following two rules to your rule set:

```
If "From" field contains "support@pmail.gen.nz", then skip next rule
If "Subject" field contains "free offer", then delete message
```

Exit When a rule triggers that has the action *exit this rule set*, all rule processing for the current message terminates at once - no more rules are examined or actioned. The primary use of this action is to separate *subroutines*, or groups of rules that you access via *call label* actions, from the main body of your rule set.

Labels A label is simply a name you can add to any line in your rule set. Labels are used by return and goto actions (see below) to transfer processing to a different location in the rule set. Labels can appear anywhere in the rule set - when calling or going to a label, you can go either forwards or backwards. Labels are simply a textual name - you can use any text or letters you wish up to 45 characters in length. Labels are the only passive item in a rule set - on their own, they do absolutely nothing, and any match conditions they contain are ignored.

Calls and returns If you have defined a label in your rule set, you can call it at any time by defining a rule with the *Call label* action. If the rule triggers, processing of the rule set will transfer to the first rule after the label you name and will continue until either there are no more rules (in which case rule processing terminates), or a rule triggers that has the *Return from call* action (in which case processing resumes at the rule following the one which initiated the call).

Gotos A goto is like a call, in that it simply transfers processing to a label anywhere else in the rule set. The difference is that you cannot return from a goto - the transfer of processing is final. Gotos are primarily useful when implementing complex logical operations.

Example
In this example, we will implement a list server where the user can subscribe to lists on your system by sending you a message containing the subject line *subscribe <listname>*: to do this, we look for the word *subscribe* in the subject line, and if we find it, we call a label that handles the list names. Note the use of expressions in this example to isolate complex cases.

```
If data matches expression "Subject: +Subscribe*" then call subscriptions
[… other rules can appear here…]
If data matches "*" then exit from this rule set

Label subscriptions
If data matches expression "*interest-list* then add sender to list "ilist"
If data matches expression "*beta-testers*" then add sender to list "beta"
[… other subscription cases could appear here…]
If data matches "*" then return from call
```

Some things to note about this example: firstly, notice the way the subject line is detected: the regular expression Subject: +Subscribe* detects a line that starts with the letters subject:, followed by one or more spaces (the + operator) then the letters subscribe. Note that the Return from call rule matches on a

single * - this is the way you tell a rule to match on any text, and in this case ensures that the return will always be triggered. Finally, notice the use of an *Exit from this rule set* action to ensure that we don't drop into the subscription processing rules other than at the proper times.

Creating logical operations in your rule sets

Often, you may only want to apply a rule to a message if all of a number of conditions are matched, or if any one or more of a number of conditions apply. This kind of operation is known as a *logical operation* (it is also known by the technical name *Boolean operation*). Mercury implements logical operations by a combination of rule order and flow control structures. Before reading about logical operations, we strongly suggest you read the section on flow control.

Tip When using logical operations in rule sets, it is very important to remember that rules are always applied to the message in the order in which they appear in the rule list editor, starting at the top of the list and working through to the bottom.

Applying a rule when any of several conditions is met (logical OR)

The simplest logical operation you can create in a rule set is that where an action is applied if one or more conditions is satisfied (i.e, condition1 OR condition2 OR condition3 and so on). You can create this kind of operation simply by creating multiple rules matching on different conditions, all executing the same action. For example, say you want to highlight a message in Magenta in your folder if the subject contains the word order, or if the subject contains the word invoice or if the To: field contains the address orders@foo.bar.com - you would add the following three rules to your rule set

```
If "Subject" field contains "order" then highlight message in magenta
If "Subject" field contains "invoice" then highlight message in magenta
If "To" field contains "orders@foo.bar.com" then highlight message in magenta
```

Notice that the action is the same in each case. In cases where repeated application of the rule action might not be desirable (for instance, copying messages to another user, in which case that user could get multiple copies of the message), more complex combinations of goto and call statements can be used to achieve the same effect - for example, like this:

```
If "Subject" field contains "order" then goto label "copy message"
If "Subject" field contains "invoice" then goto label "copy message"
If "To" field contains "orders@foo.bar.com" then goto label "copy message"
label "Next label"
[… other rules …]

Label "copy message"
If "Subject" contains "*" then copy message to "orders"
goto label "next label"
```

In this example, any of the conditions will transfer control to the rule that actually copies the message, which in turn immediately transfers control to the first rule after the group, so you will only ever get one copy of the message. Note the use of the single asterisk (*) in the copying rule: a single asterisk matches any text and in this case ensures that the rule will always trigger.

Applying a rule only if all specified conditions are met (logical AND)

Mercury offers several ways ways of applying a rule or rules only if all of a set of conditons apply. Much the easiest is to create a set of rules all of which have the *Logical AND* action; when you do this, Mercury will apply the action defined in the last rule provided all the preceding rules with the *Logical AND* action trigger properly.

There are two other ways to create logical AND operations that may be useful in some cases. The simpler form allows you to match exactly two conditions, using the *Skip next rule* action. To do this, you simply use the *Skip next rule* action on the first rule in the pair if the data does NOT match the first condition, then apply the action you want in the second rule only if the second rule DOES match the second condition. For instance, in the following example, we want to delete the message only if the subject field contains free offer and the from field contains cyberpromo.com.

```
If "Subject" field does not contain "free offer" then skip next rule
If "From" field contains "cyberpromo.com" then delete message
```

The more complex approach to matching multiple conditions depends on using a call statement to transfer to a group of rules where each rule returns if it does not contain the required text. This approach requires more setup, but allows you to match on an unlimited number of conditions. For instance, say we want to play a sound when we get a message from anyone at compuserve.com where the subject line contains the word Transylvania but not the word vampire, and the To: field contains the address foo@bar.com: to achieve this, we would create the following rules in our rule set -

```
If "From" field contains "compuserve.com" then call label next-test
[… other normal rules are here…]

Label next-test
If "Subject" field does not contain "Transylvania" then return from call
If "Subject" field contains "vampire" then return from call
If "To" field does not contain "foo@bar.com" then return from call
If "Subject" field contains "*" then play sound "tada.wav"
If "Subject" field contains "*" then return from call
```

Note the use of a single asterisk (*) in the last two rules to match any text: this ensures that the rule actions will always be applied. In order to get to the rule that plays the sound, all the rules before it must have failed to match.

Regular expressions: a *regular expression* is a description of an expression which permits matching based on patterns rather than on exact correspondence. If you have ever used a DOS command like "DEL *.BAK" then you have already used a regular expression -- what this command is really saying is "Delete all files where the name contains anything and the extension is BAK".

When you compose a regular expression, you use a combination of *literal characters* which must be present (such as the "BAK" in the command above) and special *metacharacters* (also called *wildcard characters*) which can match varying numbers of characters, or characters in specific groups only. As an example, MS-DOS supports two metacharacters - *, which means "match any number of any characters" and ?, which means "match any single character in this position".

MercuryE, Full SMTP Delivery Client Module

See also: Logging and session logging
The MercuryX Scheduling module and ETRN

Unlike the MercuryC SMTP module, which relies on using a single "smart" host to relay all outgoing mail for it, the MercuryE module is capable of delivering mail directly to the recipient's mail server. It does this by using Internet name resolution functions to work out the address of the proper system, then connects to it directly to perform delivery.

MercuryE is very easy to configure: all it needs is the address of a name server it can use to perform domain name lookups, which it will typically obtain from the Windows registry. Apart from that, its operation is largely automatic. It periodically polls the mail queue for outgoing jobs, handling them as required. A single job may require connections to multiple hosts, and individual addresses may have problems within the message - MercuryE handles all these conditions gracefully and predictably.

MercuryC or MercuryE?

The decision about which Mercury SMTP client you should use depends on a couple of factors:

1: Do you use a dialup connection? If you connect to the Internet using a dialup connection, you will normally use the MercuryC module because it minimizes the time you are connected. You can, however, use MercuryE on a dialup connection if doing so fits your needs; when using a dialup connection, you may have to tell MercuryE to use a specific name server, via its configuration dialog, rather than relying on Windows to provide one correctly.

2: Are you behind a firewall? If your local area network is protected by a firewall, you will often need to use the MercuryC module because systems outside your firewall will probably not be able to contact your system directly.

Technical support

If you have a problem with Mercury/32, the following resources are available to you.

1: Formal technical support If you have purchased a Mercury Manual Set, then you are entitled to obtain formal technical support directly from the developer of the program. Send a detailed description of your question or problem to

```
tech-support@pmail.gen.nz
```

Make sure you quote your manual license number in the **subject field** of your message. It is **very** important that you quote your license number in order to get formal support; if you do not include your license number, you will probably not get a reply of any kind to your message.

2: Our web page Copies of our Frequently-Asked-Question lists, and downloads of current utilities and versions of both Mercury and Pegasus Mail are available from our web sites,

```
http://www.pmail.gen.nz
```
or
```
http://www.pmail.com
```

3: FAQ (Frequently-Asked-Question) lists Anyone may access lists of common questions and answers concerning Mercury/32 by sending any e-mail message to

```
faq-merc32@pmail.gen.nz
```

The FAQ file will be returned to you automatically by our mail server.

4: User groups Various public mailing lists and user groups exist that offer valuable resources for Mercury users. To retrieve a list of available lists and groups, and instructions on joining them, send any e-mail message to

```
faq-usergroups@pmail.gen.nz
```

The FAQ file will be returned to you automatically by our mail server.