



Welcome to Mercury/32!

This guide will help you install and perform basic configuration of Mercury/32. Click any of the links below for more information on that topic.

[What is Mercury/32, and why do I need it?](#)

[Terms and conditions of use](#)

[Feature overview](#)

[What's new in this version of Mercury/32](#)

[Installing Mercury/32 to match your specific needs](#)

[System requirements](#)

[Installation overview](#)

[Getting technical support and manuals](#)

Mercury Mail Transport System, Copyright © 1993-2003, David Harris, all rights reserved.
Proud to be a product of New Zealand.

What is Mercury/32, and why do I need it?

See also: [The history of Mercury](#)

Mercury is a *Mail Server* - that is, a program that provides centralized mail services to a collection of related users' mailboxes on your network or computer. The program consists of a core application, which handles delivery of mail to local mailboxes, and a number of special plugin modules called *protocol modules*, each of which implements a major Internet protocol related to electronic mail. You can "pick and mix" protocol modules to produce a combination that exactly matches your mail system requirements.

The key concept in understanding what Mercury is and why you might need it is contained in the word *centralized*. Mercury is intended to be the hub for all your mail access, and as such it is most useful if you have more than one user needing access to Internet e-mail.

Using Mercury as a mail server can confer the following benefits (among others):

- * Increases the security of mailboxes on your file server, because your users do not need rights to access other users' mail.
- * Makes management easier by centralizing it - you can administer your mail users and their settings from a single point of access.
- * Allows multiple different mail clients to access the same mailbox data
- * Mail can be received even if your users' machines are turned off.
- * Allows specialized features like automatic replies, automatic forwarding and mailing list distribution without needing the users' machines to be turned on.
- * Gives you control over content - Mercury offers extensive filtering and logging, so you can get rid of unwanted "spam" before it gets delivered to your users' mailboxes, or can watch for unauthorized transmissions from inside your organization.
- * Acts as an efficient way of sending mail internally even if the recipient is away or has his or her computer turned off.
- * Allows efficient use of expensive connection services like ISDN or ADSL - only the computer actually running Mercury need have access to the outside world.
- * Frees you from having to have your ISP provide mailboxes for you.

Mercury is an ideal companion for our mail client program, *Pegasus Mail* (available from <http://www.pmail.com>) - the two programs have special integration code that allows them to work closely together. Having said that, you can use Mercury as a mail server with any applications that support standard Internet mail protocols.

What does all this cost? Nothing - nada, zippo, rien, nix... Mercury has been provided free of charge as a service since 1993. As such, it is an anachronism, dating from a time when the Internet was different (and in the opinion of the author, much more humane than it is today). We *can* sell you technical support and manuals, but these are strictly optional - you do not need to purchase anything, and we have tried to provide you with all the information you need to run the program successfully within this package. If this all sounds too wierd, then perhaps you might like to read the history section to see how it all came about, and why it carries on so wrong-mindedly by the standards of these mercenary times.

System Requirements

To run Mercury/32 you need to meet the following minimum system requirements:

- * *Operating system* Windows 95, 98, 98SE, ME, NT4 (with SP2), 2000, XP or XP Professional. Mercury/32 can also run on the server versions of Windows NT4, 2000, and XP, and on Windows Server 2003.
- * *Total system memory (RAM)* (assuming no or minimal IMAP server usage)
 - Windows 95/98/ME, 16MB
 - Windows NT4, 2000, 128MB
 - Windows XP, XP Pro and all server versions, 256MBFor IMAP usage, add 300KB per active connection.
- * *Disk space* 4MB for the Mercury program files, plus space occupied by mail and log files, as required.
- * *Novell NetWare Support* Mercury will work with NetWare 3.x, 4.x, 5.x and 6.x. To run in NetWare mode, you must be using genuine Novell workstation client software on the machine where Mercury runs: the Microsoft workstation software is insufficient.
- * *TCP/IP Networking* Any properly-installed network card or modem driver supported by the Microsoft Windows Sockets interface. Put simply, if you can view a web page in Internet Explorer on the machine, then Mercury can run there.

Terms and conditions of use - your license to use Mercury/32

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 or promotion for products or services 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, whether consequential or real, 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 ("spam").

Mercury/32 feature overview

Shakespeare said "Nothing will come of nothing", and the common peoples' wisdom usually tells you that "you get what you pay for". On these grounds, since Mercury is free software, you might expect its feature set to be limited in some way... Well, we think Mercury is one of the richest, most powerful mail processing environments you can get at any price, and we believe that you will think so too after you've tried it. Below is a partial list of the features Mercury offers.

Protocols supported

- * SMTP (server, relay-based client and full end-to-end delivery client)
- * POP3 (server and distributing client)
- * IMAP4rev1 (with multiple simultaneous access to the same mailbox)
- * PH (server, for directory lookups)
- * Finger (server, for directory lookups)
- * PopPass (server, for remote password changing)
- * HTTP (server, allows web-based mailing list subscription management)
- * SSL/TLS for encrypted data transfer over SMTP, POP3 and IMAP4

Core module features

- * *Multiple domains* Support for multiple domains on one system. A single Mercury/32 server can service a practically unlimited number of different e-mail domains.
- * *Aliases and addresses* Support for unlimited aliasing and alternative address forms (so, a user can have any address, not just his or her username@your.domain).
- * *Autoresponders* fully-programmable automatic replies, including options allowing different replies to be sent automatically depending on the day of the week, the month, the time of day or during an arbitrary range of dates. Mercury's autoreply logic has extensive checks and balances to prevent mail loops and mail storms, which are common in other systems.
- * *Forwarding* Full support for automatic forwarding of mail, including forwarding to multiple addresses.
- * *Filtering* Comprehensive mail filtering allowing an almost unlimited number of tests and actions to be performed on a message. Filters can be global, or tied to specific addresses. Filters can be applied to attachments and can remove attachments from messages.
- * *Comprehensive content control* set up exacting tests to reduce or eliminate unwanted "spam", or to monitor the content of messages passing through the server.
- * *Policies* -apply your own external tests to mail passing through the system. For example, using a policy you can add comprehensive virus scanning to your mail server at no cost by simply linking it to a personal virus scanning tool.
- * *Mailing lists* Mercury's support for subscription-based mailing lists is so strong it deserves its own section (see below for more details).
- * *Statistics* Mercury keeps detailed statistics of many aspects of system operation, and can mail these to any address at an interval you specify.
- * *Templates* All notifications (such as delivery failure messages) can be customized using a simple text editor.
- * *Multiple queues* Mercury can service queues on many servers: this is particularly useful if you are using Pegasus Mail, since it allows a single Mercury/32 system to service an almost unlimited number of Novell NetWare or Windows NT servers.
- * *Mail server* Mercury has a built-in command-driven mail server that can be used to perform list management, alter user settings and more simply by sending it an e-mail message.
- * *Easy user interface* Mercury has an attractive, easy-to-use interface that will be familiar to any Windows user - no need to fiddle around with obscure text files to do configuration.
- * *Domain mailboxes* Mercury makes it easy to create a single mailbox that receives all mail addressed to a particular domain.
- * *NetWare support* Mercury has special support for Novell NetWare local area networks, and can integrate with either NetWare NDS/eDirectory-based systems (such as NetWare 5

and 6), or older Bindery-based NetWare servers.

Scheduling module features

- * Mercury includes an "overseer" module that controls connection to the Internet. If loaded, this module can control when and how often Internet connections should occur, including separately-settable peak and off-peak settings for each day of the week (so, you can have it connect less frequently during the weekend, for example). The scheduling module integrates with other protocol modules to ensure that they are active when they need to be, and is especially well-suited to environments where the Internet connection is intermittent, such as dialup, ADSL or ISDN systems.

SMTP Server features (incoming mail)

- * Full support for RFC821/2821 transmission format, including ESMTP
- * Support for Authenticated SMTP using LOGIN, PLAIN and CRAM-MD5 authenticators
- * Exceptional range of methods for handling relay prevention
- * Transaction-level filtering allows you to apply arbitrary tests to incoming mail as it is received, including tests on the SMTP HELO command.
- * Compliance settings allow you to reject mail not conforming to any of a range of tests you can select. Very useful for eliminating address harvesters and spam before it even reaches your server's mail queue.
- * Short-term blacklisting allows clients that break too many rules or break them too often to be blacklisted on a short-term basis - once again, a powerful tool for dealing with address harvesters and relay-based spammers.
- * Killfile allows mail to be refused from certain domains or addresses; the killfile can be updated dynamically and by e-mail to the server.
- * Connection restrictions allow connections to be refused from certain addresses or address ranges.
- * Support for "spam blacklist" services such as RBL. You can set up any number of blacklist query definitions and can tag (mark with a header), redirect or reject messages that are blocked by a service. Both address-based and domain-based blacklist services are supported.
- * Full session logging capability allows you to record every character that passes between the server and the connected client.
- * Listen on multiple ports - Mercury can listen on any port you choose and can be told to listen to two different ports simultaneously if you wish (handy for working behind firewalls or with proxy servers).
- * Supports ESMTP size restrictions for controlling the size of incoming mail

SMTP Client features (outgoing mail)

- * Choose between either relaying SMTP (where Mercury connects to another computer to send mail) or full end-to-end delivery. Relayed SMTP is especially useful behind firewalls, or when you have an intermittent Internet connection.
- * Full session logging, allows you to see every character that passes between the client and the server to which it connects.

POP3 Server features (remote new mail access)

- * Connection restrictions allow connections to be selectively permitted or refused based on addresses or address ranges.
- * Special profile options allow you to tailor the behaviour of the mailbox, including options to present only unread mail, options to make deletions final, preventing deletions from the mailbox and more.
- * Individual POP3 profiles allow customization of the server's behaviour for each mailbox.

Distributing POP3 Client features

- * Can be configured to download mail from an unlimited number of remote POP3 mailboxes and to process that mail into any number of local mailboxes (so, you can have four remote

mailboxes all downloaded into one local user, or each remote mailbox being downloaded to separate users).

- * Supports "domain mailboxes" - can distribute mail properly to local users even if all mail for your domain ends up in a single mailbox. The POP3 client uses extensive heuristics to work out exactly to whom each message in a domain mailbox is actually addressed.
- * Extensive customization options allow you to tailor the behaviour of the client to suit your circumstances: for instance, you can specify non-standard headers in messages that contain recipient addresses, and you can specify a "default user" who should receive all mail that cannot be explicitly identified as local.

IMAP4rev1 Server features (remote mailbox access)

- * Practically maintenance-free - no specific per-user configuration is necessary: just turn the server on and your users can retrieve mail using any IMAP-compliant mail package.
- * Tested with many webmail packages that use IMAP as a back end, including IMP, Twig, SquirrelMail and others.
- * Allows multiple separate users to connect to the same mailbox simultaneously - especially handy for helpdesks, support lines and other situations where shared write access to folders is required.
- * Connection restrictions allow connections to be selectively permitted or refused based on addresses or address ranges

HTTP (web) Server

- * Provides mailing list subscription management over the web (other web-based services, including remote administration and webmail are in development)
- * Extremely robust - HTTP has developed a bad reputation as an insecure protocol, but Mercury's service-based model is extremely robust and resistant to hacks and exploits.
- * Maintenance free - turn it on and forget about it.

Subscriber-based Mailing List features

- * Simple e-mail based subscription and unsubscription
- * Allows both moderated (managed) and unmoderated mailing lists. For moderated lists, you can choose to allow only moderators to post to the list, or to make posting public, but to have moderators the only ones able to add or remove subscribers. A "primary moderator" can also be designated to receive rejected list postings automatically.
- * Web-based subscription management - users can subscribe to and unsubscribe from lists using a web browser, and can maintain their subscriptions using a simple web interface.
- * Member lists can be made available to subscribers or not, as required. Simple mail-based commands allow enumeration of list membership where it is permitted.
- * "Rolling archives" of all mail posted to lists can be created automatically.
- * Three levels of error control, including VERP (Variable envelope return processing) for total automation of error-handling.
- * You can create "welcome" and "farewell" messages that are automatically sent to subscribers on creation and termination of their subscription respectively.
- * Confirmed subscription: you can force users to send a special confirmation message before their subscription becomes activated.
- * Full support for automatic generation of mail digests, using the MIME digest format. Individual subscribers may choose whether or not to receive list mail in digest form. When digests are sent can be based on either digest size or elapsed time.
- * Subscription expiration - subscriptions can be configured to expire after a certain number of days.
- * Posting restrictions: lists can be configured to accept mail from anyone, from subscribers and moderators only, or from moderators only.
- * Size restrictions: lists can be configured not to distribute mail larger than a size you specify.
- * Password-controlled submission: lists can be configured to accept mail for distribution only if a specific password header is present in the message (many mail clients, such as Pegasus Mail, allow you to add custom headers to your messages). Moderators and

subscribers can have separate and different sets of passwords..

- * Full support for RFC2369 List URL header fields for automating common subscription tasks for subscribers.
- * Replies can be forced to go to the list instead of the original sender or not, as required.
- * Automatic subject line modification: you can force certain text into the Subject line of messages distributed to lists, usually to assist with mail filtering by subscribers.
- * Encryption of mail distributed to lists is possible if your subscribers use Pegasus Mail.
- * Anonymous lists - you can create lists that automatically "anonymize" the mail they send - this is especially useful for "suggestion boxes" or for subjects involving sensitive content.
- * Rich subscriber settings: subscribers can be "active", "on vacation" (i.e, their subscription automatically turns on again after a set period) "inactive" (i.e, still belong to the list and able to post to it, but not receiving distributions) or "barred" (i.e, prohibited from subscribing in future).
- * Subscriber statistics - see at a glance when a subscriber joined the list, when he or she last posted, and how many postings he or she has made to date.

... And much, much more. We haven't even touched on user interface features like consoles, activity monitors, speed search in lists, pause options for each module... There's an enormous amount in there, and the only way to grasp how comprehensive the system really is is to install it. We hope this list has whetted your appetite, though.

What's new in this version of Mercury/32?

Mercury/32 v4.0 (November 2003)

V4.0, while a long time coming, includes some of the most powerful improvements we have ever made to Mercury, and creates a significant roadmap for future versions. In particular, the MercuryB web server protocol module heralds the arrival of web-based management and services for Mercury, and the inclusion of SSL support improves security dramatically.

- * *MercuryB HTTP server*: this new protocol module implements the HTTP protocol for web-based access using a loadable service module approach to provide services. MercuryB is not intended as a general-purpose web server - it is specifically targeted at tasks specific to Mercury, such as mailing list subscription management (included in this release), remote administration and web mail (in development).
- * *SSL/TLS support* The MercuryS (SMTP server), MercuryP (POP3 server) and MercuryI (IMAP4 server) protocol modules now have comprehensive, easy-to-enable support for the SSL secure sockets protocol.
- * *Web-based mailing list subscription management* Using the new MercuryB HTTP server, mailing list subscribers can now manage their list subscriptions over the web. Subscribing, unsubscribing and managing subscriptions is now as easy as using a web browser and filling in some simple forms.
- * *Heavily improved spam filtering* The content control engine in Mercury/32 v4.0 has been heavily enhanced with new tests and a new default rule set that catches vastly more spam. We are also currently developing auto-update mechanisms that will allow spam rule sets to be updated automatically, allowing faster response to new types of spam as they arise. The content control engine now also has more options for adding headers to messages and can generate diagnostic headers indicating which rules contributed to the calculated weight for the message. Mercury's autoreply engine has also been changed so that any message getting a positive content control weight will not receive an autoreply: this prevents the server from sending autoreplies in the majority of cases.
- * *Improved HTML and encoding handling in Content Control* Mercury's content control engine has been retrofitted with improvements made in Pegasus Mail v4.12, and now only checks textual components of messages. When doing so, it now correctly applies BASE64 and Quoted-printable decoding before applying its tests, and strips HTML tags from HTML content (leaving only A, IMG and base tags for later testing). Content control now has explicit tests that check for lazy HTML (IMG tags with remote references), IFRAME tags, and excessive numbers of comments. The regular expression engine has also been extended with a number of powerful new tests.
- * *Attachment filtering* Mercury's general-purpose filtering rule engine has been enhanced with the ability to filter attachments within messages. Attachment filters can work on the filename or extension of the attachment, and can delete attachments from messages if required. The filtering engine now also has an action that can add a header of your choosing to any message as it is processed.
- * *VERP support in mailing lists* VERP (variable envelope return processing)

allows near-total automation of error processing in mailing lists. Mercury/32 now includes three modes for handling mailing list errors, two of which use VERP. For larger lists, VERP can substantially reduce the amount of effort required to keep subscriber lists up-to-date.

- * *Subscription passwords* You can now specify that a password be required for subscription to a mailing list - this eliminates casual or unwanted subscriptions to mailing lists and reduces moderator workloads.
- * *Mercury/1 IMAP Server improvements* The IMAP4 server has been heavily overhauled and is now much more robust than in previous versions. Sites using stateless clients (such as webmail packages like Twig or IMP) should notice performance improvements as well.
- * *Transaction-level filtering in MercuryS* The MercuryS SMTP server now supports filtering at the transaction level - you can apply expressions to the SMTP EHLO command and to the data of the message as it is actually received. This is an incredibly powerful feature, particularly for dealing with address harvesters and spammers who attempt to relay via your server.
- * *Short-term blacklisting* MercuryS now supports the idea of short-term blacklisting, where clients that breach too many compliance conditions or match specific transaction-level filters can be blocked from connecting to the server for a period of 30 minutes. This is a powerful way of dealing with zombie systems (computers that have been taken over by hackers or spammers) and of protecting against denial-of-service attacks.
- * *Connection control overhauled* The ability of the Mercury server modules to control which machines can connect to them based on IP address has been totally overhauled. It is now possible to specify arbitrary address ranges to which restrictions can be applied, and each restriction can have specific attributes enabled or disabled (so, for instance, you could create an "Allow" entry that permitted a system to relay mail through MercuryS but which was not exempt from transaction-level filtering).
- * *Improved console output* The various Mercury server protocol modules now produce much more detailed information on the console and in the log files to show why particular actions occurred.
- * *Progressive backoff in delivery* You can now tell Mercury to use a "progressive backoff" algorithm when calculating retries for mail jobs; this tells Mercury to start with a relatively short retry period, then use gradually longer and longer retry periods the more retries occur. This can dramatically speed the delivery of mail when one-off transient glitches occur, and cuts down queue congestion caused by messages with more significant delivery problems.

Mercury/32 v3.32 (August 2002)

Mercury/32 v3.32 is a moderate update with many fixes and corrections, and a short list of useful new capabilities. Sites using the Mercury/1 IMAP server should regard v3.32 as a *mandatory* update - the IMAP server has had a significant number of fixes and performance improvements made to it.

In terms of new capabilities...

- * *New Compliance page* in the MercuryS SMTP server allows you to reject certain types of unwanted mail before they even make it into your mail queue. Combined with content control and a small amount of filtering, these options can practically eliminate the occurrence of "spam" in your mailbox.
- * *Transcripts*: the MercuryE SMTP client can now generate delivery transcripts, providing reasonable proof that a message has been successfully delivered. To use transcripts, you must first enable them in the MercuryE configuration dialog, then add an *X-Transcript-To: <address>* header to the message you are sending (you can do this easily in Pegasus Mail on the Special page of the message editor). MercuryE will return the transcript to the address you specify.
- * *New mail server commands* allow you to add addresses to the MercuryS killfile and to force Mercury to reload its user database after a manual update.

Mercury/32 v3.31 (April 2002)

Mercury/32 v3.31 is a major release, with a number of significant new capabilities.

- * *IMAP*: MercuryI, the Mercury/32 IMAP server module, is now a standard component of the Mercury/32 release. Long in development, MercuryI allows IMAP4rev1-compatible mail clients such as Pegasus Mail, Mulberry and Outlook to access entire mailbox structures in a managed way.
- * *Policies*: This powerful new feature allows you to create your own external tasks to examine mail messages. Mercury provides support for things like attachment unpacking - all you do is put together a script, batch file or program that does the processing you need, then tell Mercury how to invoke it. As an example, it takes about 10 minutes to create a virus scanning policy to check all your mail for viruses, even if the virus scanner does not understand Internet message encodings. Policies are found on their own page in the *Mercury Core Module* configuration dialog, and there is extensive help for them in the help file.
- * *Content control*: Are you sick to death of spam? So are we. Mercury now has one of the most comprehensive content examination and control features we could put together. Using the new content control option, you can apply tests of arbitrary complexity to incoming mail and take any of a number of actions based on the results. A clever weighting system allows you to aggregate a number of points with varying levels of importance during the evaluation.
- * *Blacklist overhaul*: Mercury now allows you to create a practically unlimited number of blacklist definitions (for services such as the RBL, or ORDB), and the functionality available for testing those blacklists has been thoroughly updated.
- * *NetWare support* A new NDS-mode enabler has been included with this version: the new enabler allows you to specify an object that Mercury should try to look up in the NDS database before starting any mail transaction. This approach, while a little rough and ready, allows Mercury to work reliably in an environment where servers are occasionally unavailable.
- * *Programmable autoresponders/autoreplies* Mercury now allows you to set up multiple automatic replies and have the correct one chosen based on factors like the day of the week, the time of day or the current date.

* *Numerous bug fixes* Many, many small problems have been addressed, especially in the MercuryE end-to-end SMTP delivery module.

Installing Mercury/32 to match your specific needs

Mercury is to e-mail what chameleons are to forests - it can blend into almost any environment. In this section, we'll describe some typical scenarios encountered by people wanting to set up mail servers in their environments, and show you what combination of Mercury features and protocol modules can be used to satisfy those scenarios. To use this section, read through the scenarios until you find one that looks like it might match your needs, then click on the link next to it.

- [Scenario 1](#) Your site is permanently connected to the Internet using a high-speed link such as a T1 or T3 leased line.
- [Scenario 2](#) Your site is connected to the internet via a dialup connection, and your Internet Service Provider (ISP) receives mail on your behalf, storing it in either one POP3 mailbox per user, or a single POP3 mailbox for all your users (a so-called "domain mailbox").
- [Scenario 3](#) You are on a local area network where all Internet access is channelled through a specialized system such as a firewall or proxy server - you do not directly connect to the Internet yourself.
- [Scenario 4](#) You have an ADSL or cable modem servicing your computer or network, but your ISP won't supply static IP addressing, or implements restrictions on outgoing mail connections ("port 25 blocking").

Scenario 1: Permanent Internet Connection

This is the easiest and most natural configuration for Mercury. You will perform a normal installation, enabling the following protocols modules

MercuryS SMTP Server, to receive incoming mail

MercuryE end-to-end SMTP delivery client, to send outgoing mail

To perform this installation, you will need a properly-advertised domain name for the machine where Mercury runs (so the outside world can find you), and the address of a reliable DNS server the SMTP client can use to determine delivery addresses. You may also need to set up a special DNS entry called an *MX Entry* to allow mail servers in the outside world to find you correctly - consult your ISP or DNS administrator about this.

If you are not using Pegasus Mail, or if you are, but have remote users, you may also wish to enable the *MercuryI IMAP4rev1 server* if you want to provide remote access to user mailboxes, and/or the *MercuryP POP3 server* to provide remote access to new mail.

If you are planning to host mailing lists in Mercury, you should consider enabling the MercuryB HTTP server, which provides web-based subscription management services to mailing list subscribers.

In this environment, you typically will *not* enable the *MercuryD distributing POP3 client*, or the *MercuryX scheduling module*, and there is usually no need to use the *MercuryC relaying SMTP client* instead of MercuryE.

Scenario 2: Dialup with remote POP3 mailboxes

A commonly-used configuration, this scenario involves co-ordinating periods of Internet access with Mercury protocol modules that can operate on demand to process mail. This usually means that you will enable the following protocol modules:

MercuryX scheduling module, to control dialling and module activity

MercuryD distributing POP3 client, to retrieve and distribute the remote mail

MercuryC relaying SMTP client, to send mail via the ISP's SMTP server

To perform this installation, you will need to know the address or domain name of your ISP's POP3 server (where the mailboxes are located) and SMTP server (to which MercuryC will refer outgoing mail for handling). You will also need to know the usernames and passwords for the mailboxes that the MercuryD distributing POP3 client is going to service for you. The MercuryD module can also service so-called "domain mailboxes", where all mail addressed to any user in a specific domain is placed.

If you are not running Pegasus Mail, you may also need to enable the *MercuryS SMTP server* to allow your users to send mail, and the *MercuryP POP3 server* to allow them to retrieve it. These servers will typically only provide services to users on your local network, not to outside users because of the intermittent nature of the connection.

In this environment, the MercuryE end-to-end SMTP delivery client is typically not a suitable choice for handling your outgoing mail, because it requires more prolonged access to the Internet than the relaying version. Arrange with your ISP to use one of their SMTP servers as a mail relay.

Scenario 3: LAN with firewall or proxy server

With security becoming a major concern on the Internet, this type of configuration is becoming increasingly common. Exactly how you will proceed with this configuration depends on whether your firewall is *stateless* or *stateful*: a *stateless* firewall only allows traffic directed to specific ports on your network to pass through, regardless of how the connection was established, whereas a *stateful firewall* notices when a machine inside the protected area opens a connection and automatically opens up that port to the outside world.

If you have a stateless firewall then you will need to install the MercuryC relaying SMTP client to handle your outgoing mail, and direct it to use a machine that is either outside the firewall (typically in a so-called *DMZ region*) or else direct it to a machine that is handled as an exception by your firewall to allow mail access to the outside world (this is often the machine that runs the firewall itself). Incoming mail can typically still be handled by the MercuryS SMTP server, provided the firewall has been configured to permit traffic to port 25 to pass through.

If you have a stateful firewall then you need do nothing more than use a standard, [scenario 1](#) or [scenario 2](#) installation - the firewall will see any connections made by Mercury and will allow the port it opens to be used for the duration of the connection. This is the easiest way of securing your network, and indeed, these days the majority of firewalls are stateful.

Proxy servers There is no one solution when using a proxy server - everything depends on whether the proxy server simply echoes your data, or opens a new connection and re-transmits the data. In many cases, using a proxy server may mean nothing more than altering the port addresses used by the standard Mercury modules, although you may need to use the MercuryC relaying SMTP client to handle your outgoing mail instead of the MercuryE end-to-end delivery module.

Scenario 4: High-speed connection with restrictive ISP

In this scenario, you have a high-speed (ADSL, Cable or ISDN) Internet connection, but do not have a static IP address, or have an ISP who imposes what is known as *port blocking restrictions* - that is, refuses to allow requests to open certain types of Internet connection to pass on to the Internet. This scenario is something of a hybrid between a permanent connection ([scenario 1](#)) and a dialup connection ([scenario 2](#)).

If you do not have a static IP address, you typically cannot advertise a stable well-known endpoint address for the SMTP protocol. This means that you typically will not be able to use the MercuryS SMTP server to accept mail from the outside world - instead, you will have to arrange to have your ISP host your mailboxes on one of his systems, then use the MercuryD distributing POP3 client to download them to your system periodically. You may still be able to use the MercuryE end-to-end SMTP delivery module to handle outgoing mail, though.

If your ISP imposes port blocking restrictions then you typically will not be able to handle your own outgoing mail delivery, because the port 25 connection used by the MercuryE end-to-end SMTP delivery module is likely to be one of those blocked by your ISP. Instead, you will need to use the MercuryC relaying module and arrange with your ISP to use one of their SMTP servers as a mail relay (if they are blocking port 25 connections, then they are likely to have this facility available as a standard service).

If both these restrictions apply to your situation - that is, your ISP both enforces port blocking and also does not provide static IP addresses, then you will effectively be doing a standard dialup configuration ([scenario 2](#)).

Installation overview

Installing Mercury is a fairly simple process of answering a few questions for the setup program, after which it will copy the necessary files to your computer and create a basic working configuration for you. This section of the help file aims to help you prepare for installation by explaining what the setup program is going to ask, why it wants to know, and how you can decide what the best answers are.

Step 1: New installation, or update?

The first thing setup will ask you is whether you want to do a new installation of Mercury, or whether it should update an existing installation. If you choose the update option, all your existing settings and template files will be preserved - only the actual program files for Mercury will be updated.

If you perform an update instead of a new installation, the setup process essentially ends at this point - Setup will simply perform the necessary modifications and will then exit.

Step 2: Choosing an operating mode

If you chose a new installation at step 1, Mercury will ask you whether or not you want to install its special support for Novell NetWare local area networks. There are three choices at this point - installation in NetWare Bindery mode, installation in NetWare NDS mode, or no NetWare-specific module.

* NetWare Bindery mode is suitable for NetWare 3.x file servers and in some cases for NetWare 4 and later versions, although we do not normally recommend the use of Bindery mode on NDS-based servers.

* NetWare NDS mode is suitable for NetWare 4.x, 5.x and 6.x file servers. Some extra configuration is required using the Pegasus Mail NCONFIG utility (which is provided with Mercury/32 for this purpose). NCONFIG is used to create user mailboxes on the file server.

* No NetWare support: this option is suitable for any environment - you can even use it when you have a NetWare-based LAN if you want more specific control over mailbox location and setup. You will typically select this option if you are installing Mercury on a Windows peer-to-peer or NT-based network, or on a single computer.

Step 3: Where to install Mercury/32

The next step simply involves telling Setup where it should actually place the Mercury/32 program files. You can choose any directory for this, although normally it is best to install Mercury/32 onto the hard drive of the machine where it actually runs (as opposed to installing it on a file server): doing this can prevent problems if your file server crashes at any point while Mercury is running.

Step 4: Selecting protocol modules and SMTP client

The next two questions involve choosing the protocol modules that Mercury should activate by default when Mercury runs. You should read the section of this help file entitled [Installing Mercury/32 to match your specific needs](#) before attempting to make this choice. Note that these questions only determine which protocol modules Mercury will load by default - all the available protocol modules are installed as part of the setup process, and any that you do not choose to enable at this point in the process can be enabled at any later time using the *Protocol Modules* option on the Mercury *Configuration* menu.

Step 5: Entering basic configuration information

Setup will now ask you for either two or three pieces of configuration information:

This machine's internet domain name You should simply enter whatever internet domain name has been assigned for this machine. Problems with the Windows TCP/IP networking layer mean that Mercury cannot always work this information out reliably by itself. Mercury uses this name when it constructs addresses like the postmaster address for your system, and setup will automatically create an entry in the Mercury configuration file telling it to regard this as a "local domain" - that is, one where Mercury is the point of final delivery.

Username for postmaster All mail-enabled Internet systems must have a special reserved user called *postmaster*: rather than forcing you to create a user called postmaster, Mercury simply treats the address as a specialized alias for a user on your system. Enter here the name of the user to whom postmaster mail should be directed. If you are running in NetWare NDS or Bindery mode, then this is the full username of the user (in NDS mode, you must express the username relative to the root of your NDS tree in the standard dotted notation). In non-NetWare mode, this is any valid mailbox name you subsequently create within Mercury or in Pegasus Mail. The user need not actually exist at setup time.

If you chose the MercuryC relay-based SMTP client at step 4, setup will also ask you for the name of a host to which MercuryC should connect when sending mail: enter the full domain name of the machine running the SMTP server that is going to send mail on your behalf (you can enter an IP address here as well, if you wish - this is sometimes necessary if you are installing Mercury behind a firewall or proxy server).

Step 6: Selecting a relaying mode

If you chose to load the MercuryS SMTP server at step 4, setup will now ask you what default relaying protections you want to enable. *Relaying* refers to the practice where a mail server accepts and forwards mail that is not addressed to any local users on its system. When you send a message using an SMTP-based mail client like Pegasus Mail or Eudora, you are *relaying* mail. Unfortunately, relaying has been abused in recent years by people sending unsolicited commercial e-mail, and most sites now find it necessary to place more or less strict controls on how their server may be used for relaying.

There are three possible answers to this question:

No relaying controls In this mode, MercuryS will act as an open relay. We strongly advise against selecting this mode unless Mercury is only operating on an intranet. It is considered very bad form to have an open relay on a mail server that is actually open to the Internet proper.

Normal relaying controls In this mode, Mercury will only relay mail if the sender's address is a valid local address on the system other than "postmaster". This mode prevents the majority of relaying abuse, but requires no ongoing maintenance. We recommend this mode in most cases.

Strict relaying controls In this mode, Mercury will only relay mail if the sender's e-mail address is a valid local address and the IP address of the machine from which the message is being sent has been explicitly approved by the system administrator using a connection control entry. This mode practically eliminates all possibility of relaying abuse, but requires more ongoing maintenance, especially if you have roving or remote users.

Step 7: Choosing a mail queue location

Mercury uses a directory on your computer to store mail as it processes it. The queue is also typically the location where local e-mail clients like Pegasus Mail can place mail for processing by Mercury.

If you are using Pegasus Mail, then you will typically enter a queue directory that exists on a shared volume accessible by your Pegasus Mail users. This will allow their copies of Pegasus Mail to submit messages to Mercury for processing simply by placing a file in the queue directory.

If you are not using Pegasus Mail, then it is usually best to enter a directory on the local hard drive of the

computer where Mercury is being installed: this provides the best performance and some extra security.

Note that if you wish, you can install the queue on the local hard disk, then later tell Mercury to retrieve mail submitted by your Pegasus Mail users from a *secondary queue*. This can offer some extra protection from server crashes or network downtime, since Mercury will still be able to accept incoming mail during the outage. Please consult the main Mercury help file for more information on setting up secondary queues.

.... And finally, it's show time.

That's it. If you've successfully navigated the setup process to this point, all you need to do is click the Install Mercury/32 button and Setup will do the rest. After the program is installed, run Mercury by selecting its icon in the Windows *Start* menu, and explore the program's Configuration menu for fine-tuning options. And remember, Mercury has comprehensive online help - if in doubt, look on the *Help* menu for assistance.

Getting manuals and technical support

Manuals and technical support for Mercury are available for sale as an option - and we want to stress the word "option" in that sentence - you are under no obligation to purchase anything from us; Mercury is provided first and foremost as a service.

If, however, you want to purchase manuals and support, you can do so by visiting our web site and going to the ordering page, http://www.pmail.com/mu_manuials.htm; manuals and support for Mercury are only available as part of our full support subscription agreements, which cover both Mercury and Pegasus Mail. The annual price for these agreements depends on the number of mailboxes Mercury is servicing for you, according to the following table:

1.5 users	US\$110 per annum
6-20 users	US\$195 per annum
21-100 users	US\$325 per annum
101-500 users	US\$550 per annum
501 - 1000 users	US\$800 per annum
1001+ users	US\$1000 per annum

The web page contains a link to a secure online ordering facility where you can order whichever option suits your needs. We are happy to accept company or institutional purchase orders on normal credit terms (nett 30 days).

Donations and reduced pricing: we have tried to make our pricing as reasonable as possible, but we're more interested in helping than in profiting. We like to donate manuals and support to non-profit organizations that work to preserve the environment, work for the benefit of animals, or are directly involved in improving conditions in the developing world. We're also open to being approached for reduced pricing by anyone who believes they can make a reasonable case for it. If you would like to discuss this further, please contact the author directly, at David.Harris@pmail.gen.nz.

The history of Mercury (and Pegasus Mail)

In 1989, I wrote an e-mail application in my spare time, because I needed it at my work. The people at my work seemed to like it, so I released it on the Internet because I thought other people might benefit from it too: by 1993 it was one of the most widely-used e-mail applications in the world - and I'm still as surprised about that today as I was back then. That program is called Pegasus Mail, and it is now the oldest PC-based mail system still in wide-spread use. By 1993, though, it was clear that the Internet was going to dominate the future of computing, so I sat down and started writing the Internet mail server package that I felt Pegasus Mail needed as a complement. By August 1993, Mercury was an established product, and was handling millions of mail messages every day for people all around the world.

My name is David Harris -- I'm the person who develops both Pegasus Mail and Mercury. There is no anonymous corporation behind the program, and the same pair of hands that wrote the first versions way back then is still writing them in 2003. People regularly ask me why I originally wrote these programs, and why I still make them available for free: Pegasus Mail and Mercury are closely - or even inextricably - intertwined in history and pedigree, so it's probably easiest if I describe how Pegasus Mail came about - because in doing that, I'll also expose the motivations that underly Mercury and explain why it's so important to me that it be available for free.

In 1989, the University where I worked (in Dunedin, New Zealand) installed its first Novell NetWare network. It wasn't until after we installed it that we found that it didn't include an e-mail system, but we'd already used up our budget and the commercial mail packages that were available were very expensive. To fill the gap, I wrote a simple e-mail program in my own time and made it available on the network: I was quite surprised to find that people liked it.

Early in 1990, after tidying it up a little, I made it available on the Internet at a friend's FTP site in Hawaii, expecting that four or five other sites might find a use for it... In the first week of availability, it was downloaded more than 100 times, which also surprised me. I found that I was receiving mail from people thanking me for giving them something they couldn't have afforded any other way -- communication. I grew to understand that communication had to be regarded as a right, not as a privilege: it seemed to me in 1989, as it still seems to me now, that *freedom of speech is useless if nobody can hear you*. Giving away Pegasus Mail seemed to be a means by which I could try to make communication more accessible to a much wider range of people who needed it - it seemed like a small act that might help in some minor way to level the playing field between the wealthy corporations and the rest of us.

From that time, I began a curious double existence, working at the University by day, and working on Pegasus Mail at night, refining and tuning it to add the things people were asking for. With each release of the program, usage grew, until by 1993, the demands it was placing on my time were so great that I had to make a choice between my safe University job and going out full-time to support Pegasus Mail and (by that stage) Mercury. Leaving the University gave me what I needed most -- time -- but took away what I needed to survive -- my salary.

This put me in an awkward situation: the ideals that had motivated me to make the program available in the first place were still just as valid as ever, but I also had to eat. I hit upon the idea of making the manuals available for sale as an option to support the development of the program. This allowed the software to remain free, and the addition of extensive online help ensured that the program remained useful even without the manuals: so, the larder was stocked without compromising the ideals. To this day, my only source of income remains the sale of manuals for Pegasus Mail and Mercury.

Since 1990, the world has changed: the Internet has become more or less a commodity, and people's expectations of software have altered enormously. I've worked hard to try to keep up with the expectations of my user base and to keep offering programs that fit all their needs. I enjoy making Pegasus Mail and Mercury available on these terms, knowing that they both help people: your support is a key component of making this all happen, whether it's by purchasing manuals, or by showing the programs to people who might benefit from using them, or simply by enjoying the fruits of my labour. With

your support and backing, I look forward to being able to offer both Mercury and Pegasus Mail in the future for as long as the ideals that originally drove me to write them remain relevant.

Cheers!

-- David Harris --

Author, Pegasus Mail and Mercury

Dunedin, New Zealand, November 2003.

David.Harris@pmail.gen.nz

