

Mailtraq

User Guide

Trademarks

Mailtraq is a trademark of Fastraq Limited. Microsoft, Windows, the Windows logo and Windows NT are registered trademarks of the Microsoft Corporation. Netscape and Navigator are trademarks of the Netscape Communications Corporation.

Conditions of Use

Mailtraq is supplied with a license, and may only be used in accordance with that license, or a written agreement from Fastraq Limited. You may install the Mailtraq Software and make a single backup of the software, but the backup copy may not be used simultaneously with the installed Mailtraq software. The Mailtraq Software is protected by International Copyright Law.

Copyright ©1996, 1998 Fastraq Limited. All Rights Reserved.

Credits

Project Manager/Developer	Elric Pedder
Documentation Editors	Daniel Lyon, Janet Brink
On-Line Documentatin	Jim Hill
Original Artwork	Marilyn Heeger

Thanks

We would like to thank the following people for their valuable contribution to the Mailtraq project :—

Alan English	David Zymous	Marcel Moens	Richard Sobey
Allan Reynolds	Duncan Dewar	Marco Imperi	Richard Townend
Andrew McKay	Edmund Broadley	Mark Derricutt	Roelf Renkema
Ann-Marie Joiner	Edward Betts	Mark Hill	Ross Nicol
Anne Pollard	Edward Laauwen	Mark Preston	Russ Brindley
Barry Salter	Emma Kane	Mark Ruys	Simon Fell
Ben Smithurst	Frank Murray	Marnix Keesmaat	Simon Quill
Ben Vincent	Geoff Percival	Marton Clayton	Simon Plant
Bob North	George Poulson	Martin Kerr	Simon Steele
Brenton Milne	Hugh Spencer	Martin Green	Stephen Tyers
Brian Orpin	Ian Thomas	Matthew Preston	Steve Lee
Carl Whalley	Jason Coombes	Michael Humberston	Steve Musitano
Chris Coward	Jasper Metselaar	Mike Collins	Steven Wenham
Chris Malme	Jim Hill	Mike Tozer	Stijn Bokhove
Christopher Bone	John Blackmore	Mike Vybiral	Thijs Kinkhorst
Christopher Partington	John Cooper	Mike Ziemann	Trevor Gentry
Colin Kelly	John Deakin	Mitch Bromage	Tom Martens
Colin Mansell	John Gorst	Nathan Strom	Tony Bullock
Darren Meldrum	John Raynor	Ralph Bednarski	Tony Olsen
Dave Wheeler	Jonathan Perkins	Rashied Ilahibaks	Tony Rickard
David Barfoot	Karl Prince	Richard Mein	Zac Sitzer
David Hughes	Lance Boehm	Richard Morris	
David Symons	Lindy Beveridge	Richard Ramshaw	

Table of Contents

- i. Preface vii
- 1. Introduction 1
 - Using this Manual 1
 - What is Mailtraq? 4
- 2. Installing Mailtraq 15
 - Essential Configuration 15
 - The Installation Wizard 16
- 3. Quick Start Guide 27
 - Multi-User Networks. 28
 - Creating Mail Slots 31
- 4. The Console. 39
 - Accessing the Console 40
 - Overview of the Console 42
 - The Console Objects. 45

5. Internet Dial-up	53
Configuring the Internet Connection Type.	54
Dial-Up Networking	57
Manual Dial-Up	59
Scheduled Dial-Up	60
Dial-Up Schedule Options	66
Automatic Schedules	68
 6. Introducing E-Mail	 69
What is E-Mail?	70
Message Envelopes	71
Message Headers	72
Simple Mail Transfer Protocol.	74
Mail Routing and Mail Exchanges	76
The Post Office Protocol	78
Choosing between SMTP and POP3	78
Mail Gateways	81
 7. Mail Slots	 83
An Overview of the Mail Slots	84
The Post Master	85
Mailbox Aliases	85
Mailboxes.	87

Mailing Lists and Archives	91
Accounting and Mail Groups	92
8. Mail Delivery	95
Overview of Mail Delivery	96
The Inbound Router	97
The Outbound Router.	109
The SMTP Service	112
The POP3 Service	114
Remote POP3 Collection	114
9. Administration Topics	121
Managing Users	122
Managing Mail Slots	125
Event Logging	128
Event Reports	131
The Mailtraq Database	133
10. The News System	137
Introduction to Internet News	138
News Collection	139
News and Mail Gateways	144
The Mailtraq News Server	150

11. Mailing Lists	153
What is a Mailing List?	154
Mailing Lists in Mailtraq	155
Subscription by Mail	161
Message Templates	168
Public Archives	169
 12. The World Wide Web	 175
What is the Web?	176
The Proxy Server	178
Web Crawlers.	180
The Web Server.	183
CGI and Templates	185
 13. The Web Administrator	 189
Accessing the Web Administrator.	190
User Access and Control	191
The Directory.	192
Reading Mail	194
Configuring Mail Slots	195
Accessing Web Sites	212
System Administration	214

14. Network Services.	221
The Service Manager	222
Firewalls	224
The Mailtraq Services	226
15. Scripting	235
What is Scripting?	236
What do you need to know?	237
The Scripting Language	237
Working with Scripts	245
Using Scripts	246
16. Templates	253
Introduction to Templates	254
Using Templates	255
Trigger Messages	256
Configuring Templates	258
Web Page Templates	259
A Web Database	261
17. Scripting Function Reference	265
Language Constructs	266
File Handling Functions	271

Message Handling Functions	277
String Manipulation	285
Database Functions	294
Mailtraq System Access	297
External Programs	306
18. Glossary	313
19. Bibliography	319
Index	323



Preface

E-mail may seem a relatively new technology, when compared to other forms of communication, but many are surprised to discover that it has been in use for over twenty-five years. During early experiments with networking technology, electronic messaging was one of the first useful innovations to be introduced.

For many years e-mail was confined to academic institutions, whose large scale networks were an ideal medium to develop the technology. When the number of institutions implementing networks grew, the concept of ‘internetworking’ began to evolve. Clearly, an open networking standard was necessary to allow different networks to connect to one another and the answer was the Internet Protocol (TCP/IP). The institution at the heart of this technology was the U.S. Defense Advanced Research

Projects Agency. The first internetwork was consequently called ARPANET. More recently, as the network's military origins were forgotten, it has become known as the Internet.

Recently the Internet has become available to the public through Internet Service Providers (ISP). These are companies whose networks form part of the Internet, and who can provide access to the Internet. Customers usually connect to their service providers using the telephone network, either with a permanently connected leased line, or a dialup line such as ISDN or the ordinary household analogue system.

The majority of Internet users connect to their ISP through a temporary dialup connection. They pay their service provider for their account, the services provided with it and the telephone company for using the telephone line. This is the cause of most of the problems related to the Internet today. Firstly, telephone connections were designed for carrying voice and not for digital data. Although huge advances have been achieved recently with modems, they are still only able to carry a relatively small amount of information. Secondly, while data is being carried, someone has to pay for the telephone time (although this varies considerably from region to region). And finally, temporary connections require additional management and administration.

Mailtraq was developed specifically to address these three issues. Mailtraq provides a very comprehensive range of messaging services. It has also been designed to make the most efficient and cost effective use of an Internet connection. Another important design concept of Mailtraq is how it is actually used. Many of those who have contributed to the

development of Mailtraq have observed just how complex server software can be to install, operate and maintain, and we have put a great deal of work into building the most intuitive interface possible.

We hope that you will find working with Mailtraq an enjoyable experience.

Thank you for choosing Mailtraq

The Development Team

1

Introduction

This section deals with the basics of Mailtraq and what it does. It does not go into either E-mail or *News* in depth as these will be covered in subsequent chapters. However, it does provide some necessary background to Mailtraq's many services.

Using this Manual

This User Guide has been written for both those with or without experience in administering a messaging server. If you have managed another mail server previously you will notice that Mailtraq includes many more features in one package. We shall outline these in this guide.

Unlike many other software packages our user manual is not simply a repetition of our on-line documentation. This guide concentrates on providing an overview of how Mailtraq's various processes work and how to make the best use of them.

For step-by-step information the on-line help may prove more useful. The on-line help is also intended to explain the user interface which is covered only briefly in this guide.

Some Assumptions

To simplify the manual we shall assume that you are an administrator for an organisation with several machines and that you use a modem to periodically connect to your ISP (Internet Service Provider). Don't worry if this does not match your situation. Mailtraq can be used in a variety of environments from a home user with one machine to a large network with a permanently leased line.

All about Enderson

To help illustrate the use of Mailtraq we have invented a fictitious company called "Enderson Enterprises". "Enderson's" is a small product distribution agency with about 30 employees. They run Mailtraq on their network server that connects them to an Internet Service Provider using a modem. Their service provider is called "ISP".

The topics covered in this guide

- *Installing Mailtraq* — This chapter covers the installation of the Mailtraq Server software and the use of the Installation Wizard to assist with the most essential configuration.
- *Quick Start Guide* — This chapter covers a number of basic tasks such as creating mailboxes, mailing lists and user accounts.

- *The Console* — This chapter introduces and describes the structure of the primary Mailtraq user interface and how to use it.
- *Internet Dial-Up* — This chapter covers the use of Dial-Up Networking service to connect to the Internet and how to use Mailtraq to automate this procedure.
- *Introducing E-Mail* — This chapter introduces the concept of electronic messaging, covering how messages are structured and stored.
- *Mail Slots* — This chapter introduces the Mailtraq concept of mail slots (including mailboxes, mailing lists and archives), which are the ‘recipients’ of mail in a Mailtraq system.
- *Mail Delivery* — This chapter covers the sending of e-mail messages to and from Mailtraq and within Mailtraq itself.
- *Administration Topics* — This chapter introduces the user accounts and how mail slots are created and managed. This chapter also covers various other administrative subjects such as event logging and reporting.
- *The News System* — This chapter introduces the Internet News system and how Mailtraq uses it. This chapter also covers Mailtraq’s own news services and the integration between e-mail and news.
- *Mailing Lists* — This chapter covers the Mailtraq mailing lists, including how to administrate and automate them.

- *The World Wide Web* — This chapter introduces the web, and describes Mailtraq's own web services (including the Web Proxy).
- *The Web Administrator* — This chapter follows the web services with the introduction and use of Mailtraq's Web based remote administration system.
- *Network Services* — This chapter introduces the Mailtraq network services and covers in detail those services not discussed elsewhere in this guide (including the TCP/IP proxy tunnel and mail gateway services).
- *Scripting* — This chapter introduces and defines the Mailtraq scripting language and includes a number of script examples and how to automate tasks with the system.
- *Templates* — This chapter covers the automatically generated e-mail messages and how to customise and extend them with the scripting language.
- *Scripting Function Reference* — This chapter is a complete reference for every scripting function provided by the standard Mailtraq function library.

What is Mailtraq?

Mailtraq is a *Messaging Server*. It sends, receives and processes messages, both *E-Mail* and *News*.

Mailtraq deals with two types of messaging: *E-Mail* and *News*. E-Mail is the well-known mechanism which is very close to its traditional counterpart: ordinary mail. The only real difference is that e-mail is nothing but digital information. Just as with mail, the messages are posted by you, and then carried around the Internet by electronic postmen and delivery vans, eventually finding their way to the intended recipient's mailbox.

As with ordinary mail, the messages are posted and then passed around the Internet from one service to another, eventually being delivered to the recipient, whereas news is simply posted on a notice board for anyone to read. E-Mail and news are exactly the same with the exception of the delivery mechanism. However, Mailtraq's Mailing Lists enable messages to be distributed to any number of recipients and the list of recipients can be managed by Mailtraq.

What gap there is between these two types of messaging is bridged with Mailing Lists. As with News articles, the eventual readers are not necessarily known to the author. However, like e-mail, each message has a specific list of recipients to whom it is delivered (they can, in turn, pass it on to others).

Mailtraq also provides a number of World Wide Web facilities, reducing the need for additional software, and making it possible to integrate them with the messaging services.

Clients and Servers

Mailtraq provides a number of 'services' available to all machines to which it is connected, whether on the local network or the Internet.

Mailtraq provides a number of 'services', which are available to both the machine on which it is running and other machines on the network and Internet. Several services can be provided at the same time, and several instances of each service can be active at once. The objects that connect to services are called clients. For example, if a user has written an e-mail message, and they press the 'send' button, it is sent by the e-mail software (the Client) to the e-mail server. In this case, the service being provided is called SMTP (Simple Mail Transfer Protocol)¹. Several people could be sending messages at the same time, so several clients could be connected at once.

While all this is happening, people can also be collecting mail from their mailboxes using another service. Mailtraq itself also connects to services provided by other servers (for example, to forward a message to be delivered to someone elsewhere on the Internet). Mailtraq is built around this concept of clients and services, and you will find out more about these later.

Where does the mail go?

When you send a message from your e-mail software it disappears off to a server which then carries it away (eventually to its intended destination). But how do you read new mail? Most e-mail software is also able to 'collect' mail for the user (sometimes sending and fetching mail take place at the same time). When the user presses the 'collect' button, the e-mail software connects to the server and fetches the e-mail waiting for it.

¹ Many Windows e-mail clients can send messages via MAPI (Microsoft's Messaging Technology). This is not currently supported by Mailtraq.

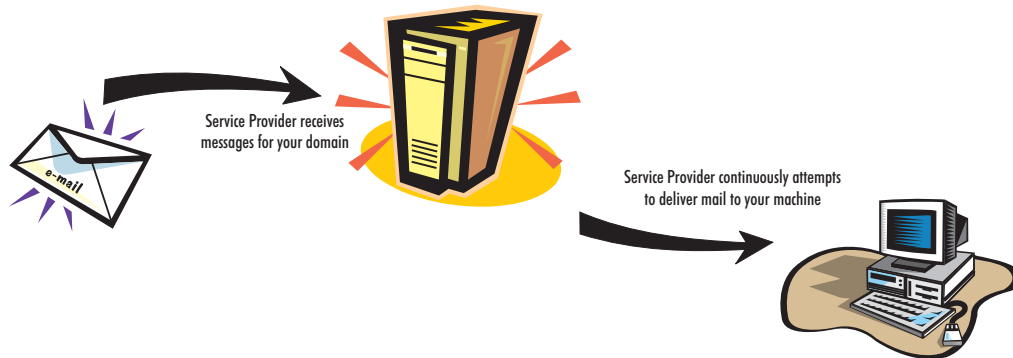
You may now realise that there are two distinct types of delivery taking place here: sending mail, and collecting mail. Usually, when you send a message to somebody elsewhere on the Internet, the message is delivered by a number of e-mail servers (these deliveries are called 'hops'). For example, you might send the message to Mailtraq, which then holds it for a period before sending it to the e-mail server at your Internet Service Provider. It will hold it before connecting to the recipient's service provider and sending it to their e-mail server. The last stage in the delivery process is to store the message in some location from which the recipient can fetch it.

Your Internet Service Provider may store the messages in one or more mailboxes for you, or it may send it to you and expect your system to store it for your users. Some Service Providers support both of these types of delivery. Mailtraq also supports both types, so it is simply a matter of configuration.

Delivering Service Providers

These Service Providers will describe this as delivering mail to you via SMTP (Simple Mail Transfer Protocol). This protocol is a method for sending (not collecting) mail. The distinction is that the delivery is initiated by the sender. Normally, the Service Provider will wait for you to connect to them and then keep trying to send the message until it is successful. While you are not connected, it simply stores the message somewhere. Some Service Providers need to be told when it should start sending mail, and Mailtraq does provide a way to do this, but you should refer to your Service Provider for instructions.

This method of mail transfer is the same as the method by which you send mail using your e-mail software (except for mail software that uses directory gateways, and MAPI clients).

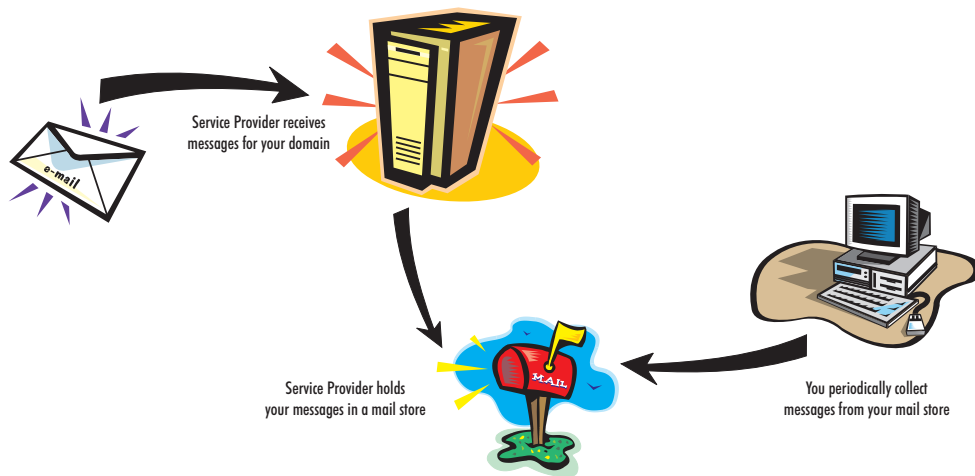


Service Provider delivers to your machine

Mailbox Providers

The Service Providers that supply mailboxes from which you collect mail need special consideration. In this case Mailtraq must be configured to connect to them and fetch new mail. There are many advantages to this method, but there are also many disadvantages. Because Mailtraq initiates the delivery, it is in control, and decisions can be made about what to download. As you will see later, this can be very useful.

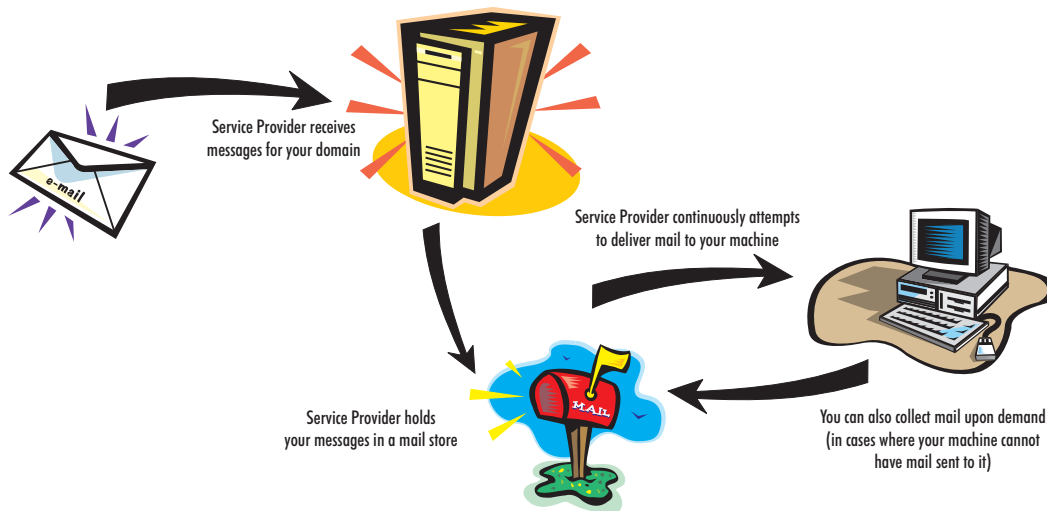
Unfortunately, this method of delivery does not scale well when one mailbox has to be used for more than one person. The obvious problem is deciding who gets which messages. Mailtraq has a number of ways to deal with this issue, but again it's a matter of careful consideration.



Service Provider holds your messages in a mail store for later collection

Dual Providers

Some Service Providers offer both methods of mail delivery. This is clearly the best of both worlds, but a new problem is created. As mentioned above, the method of delivering mail (SMTP) is the same for the Internet Service Provider as for your machines' e-mail software. Remember also that SMTP is initiated by the sender. If you wish to collect mail (instead of receiving it) from your service provider, how do you stop it sending the mail anyway? This is done by deciding which machines can use the service, and denying that service to your service provider's machines. This is handled most effectively through a firewall (discussed on page 192).



Service Provider attempts to deliver messages directly, but also stores messages for collection (as an alternative)

Internet E-Mail Addresses

You are probably quite familiar with e-mail addresses. They are in the form `user@domain`. If you think of the e-mail address as being the address on an ordinary envelope, the *user* part is the name of the recipient, and the *domain* part is the geographical address.

The Postman who is delivering your letter doesn't know anything about the person mentioned on the envelope, but he probably knows how to find the address. Once the letter gets pushed through the door, it is up to the people inside (who hopefully know the recipient) to select who will read it.

This is similar to how mail delivery works on the Internet. The 'domains' are the places on the Internet to which mail can be delivered, and each domain is responsible for getting messages to the people there.

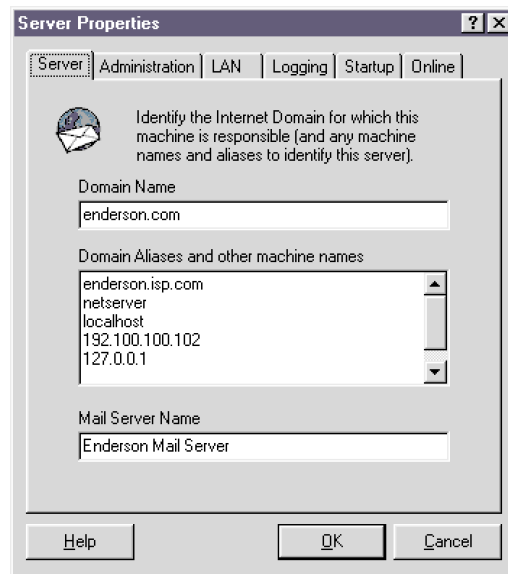
Depending on your organisation, and your Internet Service Provider, you may be responsible for an entire domain, or perhaps several domains.

For example, Enderson Enterprises may have obtained the domain name *enderson.com* from their service provider. Thus, they will be expected to receive mail for people *@enderson.com* — for example, *john@enderson.com*.

Mailtraq will need to know your domain name to determine which messages it is responsible for. It will also need to know about all the other domains for which it is responsible. For example, the service provider may actually provide an account called *enderson.isp.com*, and Enderson may have purchased the name *enderson.com* in addition to that account. They will therefore receive mail addressed to either. Further, they may have other names used only within their organisation (for example, a separate name for each machine).

Until Mailtraq has been given these names, it won't know what to do with any mail it receives. These names can be entered when asked by the Installation Wizard, and they can be set in the Server Properties (available from the *Options* menu in the Mailtraq Console).

Setting the Local Domain and Aliases



You may not actually have an Internet domain name at all. For example, your service provider may give you just one or more mailboxes, with addresses such as *enderson@isp.com*. Clearly, you are not responsible for all mail addressed to *isp.com*. In this case, you should provide Mailtraq with some domain name that is only used within your network (such as *enderson.isp.com*), and simply make sure that mail addressed to *enderson@isp.com* is dealt with as a special case.

Telling Mailtraq about how to recognise mail for your organisation is practically the only piece of essential configuration you need to do. Everything else you can configure as you go along. All the remaining issues relating to e-mail will be discussed in Chapter 6.

Getting On Line

If you don't have a permanent telephone connection to your Internet Service Provider, you will need some way of connecting to them. The Internet is a huge web of telephone lines and network cables connecting

the many thousands of machines together. In order for you to be a part of this web, you simply need one connection between your network, and a machine that is already a part of this web. In most cases, the part of the Internet that you connect to will be your Internet Service Provider (from now on, your ISP). In most cases, you connect by having a machine on your network dial a telephone number provided by your ISP. Although you can have another piece of software actually initiate this connection, most users let Mailtraq handle this.

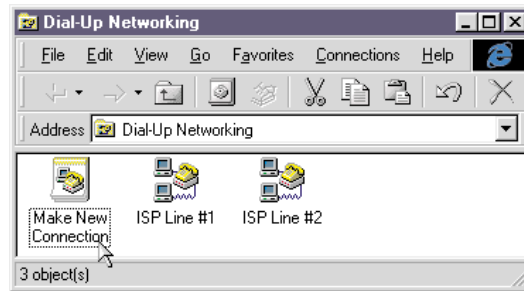
Mailtraq connects to the Internet using Windows Remote Access Service (RAS). In both Windows '95 and Windows NT this is often called Dial Up Networking (DUN). The two terms are interchangeable (at least from Mailtraq's perspective). For simplicity, we'll refer to this as DUN throughout this document.

Since we are discussing acronyms, another one to be aware of is TAPI (Telephony Application Programming Interface). TAPI is the set of communication services that include RAS/DUN, and it is used to ensure that different software applications can share telephone related devices. Mailtraq is TAPI compliant, in the sense that it only uses TAPI services (specifically DUN) with regard to your modem.

If you are using other modem software which is not TAPI compliant, then you may experience problems sharing the modem between Mailtraq and the other software.

One solution is to provide a separate modem for that software, or run it on a different computer.

Configuring Dial-Up Networking



Your ISP will have provided details for configuring a DUN connection, and you should have already done this before configuring Mailtraq. By now, you should have at least one entry in your DUN (or RAS) phone book that will connect your machine to your ISP. All you need to do is tell Mailtraq which of your phone book entries it can use.

You can still create a dial-up connection in the Install Wizard

You will have been asked to select one entry in the Installation Wizard, but you can configure others from the Dialup Properties (from the *Options* menu in the Mailtraq Console). For each entry, you must also provide an account name and password (even if you have already entered these elsewhere).

Whenever Mailtraq needs to go online, it will simply initiate one of these connections. Additionally, if it detects that one of the connections is active, then it will make use of it.

2

Installing Mailtraq

This chapter covers the installation of the Mailtraq server, and the most essential configuration. Mailtraq helps users get started by providing an Installation Wizard to assist with the most important configuration options, providing step-by-step instruction, and ensuring that the basic configuration settings will be effective.

Essential Configuration

As mentioned in the introduction, in order to get Mailtraq ‘up and running’, and carrying messages safely, there are actually very few things that must be configured. Everything else can be refined over time while

the system is actually in use. The Installation Wizard makes sure you give Mailtraq the most vital details and provides enough configuration to get the system operational with as little effort as possible.

This chapter will concentrate on the Installation Wizard. You should refer to the following chapters for help with the other systems configuration.

You don't have to use the Installation Wizard at all if you do not wish to do so, but you must ensure that the issues discussed in the previous chapter are configured before you begin using Mailtraq. If you skip the Installation Wizard, the default settings will be applied.

The Installation Wizard

Instructions on how to begin the installation process can be found with the distribution CD.

The host machine is the machine on which Mailtraq is to be run

The first stage of the installation process is to install Mailtraq on to the host machine. This involves steps such as entering the registration details, selecting a destination directory, and selecting the components of Mailtraq to be installed.

The second stage of installation involves the initial Mailtraq configuration. The Installation Wizard can assist by asking a series of questions, and then performing the essential configuration based upon the data you enter.

Remember that you can change any aspect of the Mailtraq configuration through the console after you have completed the installation. You will not be tied to the settings you enter in the Installation Wizard.

Internet Service Provider Profiles

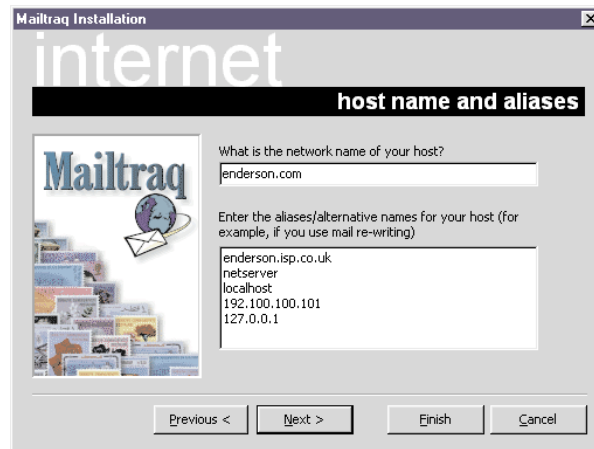
The Installation Wizard has profiles of many Internet Service Providers. These profiles supply Mailtraq with information such as the names of mail hosts, web cache services and domain name servers.

If your service provider is not listed, you should select the 'Not Listed' entry, rather than a similar service provider. Your ISP's technical support service will also be able to provide this information.

Domain Name and Aliases

This stage of the Installation Wizard involves setting the Domain name that Mailtraq will manage. There are two purposes for this. Firstly, Mailtraq must be able to recognise the mail that is its responsibility. Any e-mail address that Mailtraq does not recognise will be sent to another mail server.

Secondly, Mailtraq must be able to sign messages it generates with a domain that other mail servers will be able to recognise (should return messages need to be sent).



Mailtraq will take responsibility for any e-mail domain that matches the primary domain and any aliases. Mailtraq will only use the primary Domain name when creating it's own messages.

- If your account with your service provider does not include a domain name (i.e. they supply only one or more POP3 mailboxes under their own domain), then you should define your own unique domain name.

Mailtraq also uses these names to identify local web page requests, so you should enter any name that represents the server machine on your network.

It is worthwhile entering the IP addresses by which the server machine is recognised on your network too, as it is possible to address mail and access web sites based on the IP address.

Outbound Mail Delivery

This stage allows you to enter the mail servers (normally those of your Internet Service Provider) to which you can send outbound e-mail.

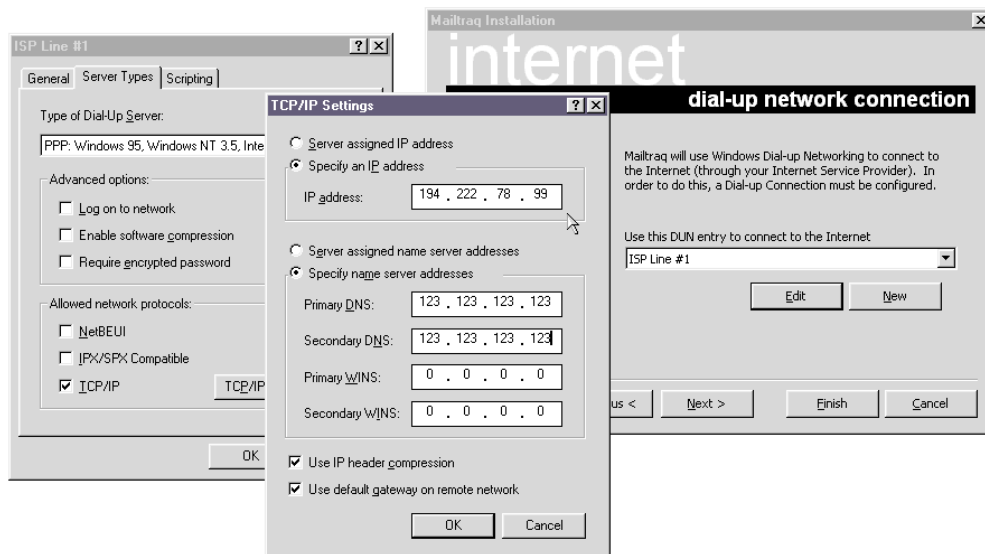
These are called 'smart hosts', because they will accept mail to any destination, and take the responsibility of routing it themselves.



Mailtraq can also route outbound mail itself (identifying the most appropriate server for each message). However, there are times when this is inefficient, and occasionally Mailtraq will be unable to resolve the route itself. For this reason it is important to enter a 'smart host', even if you intend to use Mailtraq's routing facilities.

Dial-up Network Connection

Mailtraq will need at least one dial-up connection to be configured in Windows in order for it to connect to your Internet Service Provider (to deliver and receive mail).



Your service provider should have supplied details on how to create a DUN connection. If they have not, you will need the following information from them :—

- Whether or not you have a static IP address
- If so, what that IP address is
- The IP address of your service provider's DNS machine
- A dial-up telephone number
- Your dial-up account name and password

This information can be used to create a Dial-up Network Connection. If you encounter difficulties using the connection, you should contact your service provider for assistance.

Service Provider Account Details

In this section, you should enter the account name and password that you use to connect to your Internet Service Provider. Mailtraq needs this information even if you have already entered it as part of your dial-up configuration.

Create a User Mailbox

This section allows you to (optionally) create an initial mailbox. If you are configuring Mailtraq for a single user environment, then this is all the configuration you will need for your mail-slots.



The image shows a screenshot of the 'Mailtraq Installation' window, specifically the 'configure the mailbox' step. The window has a title bar that says 'Mailtraq Installation' and a close button. Below the title bar, the word 'mail' is displayed in large, light blue letters. To the left of the input fields is a Mailtraq logo featuring a globe and a stack of mail slots. The main area contains three input fields: 'Name of the mailbox owner (Or other description)' with the text 'John Smith', 'Mailbox name (the email address)' with the text 'johns', and 'Password (for access via POP3 from a mail client)' with a masked password '*****'. Below these fields is a note: 'One can post to this mailbox from other hosts, and access this mailbox over the network from any mail client.' At the bottom of the window are four buttons: 'Previous <', 'Next >', 'Finish', and 'Cancel'.

However, if you are installing Mailtraq to manage e-mail for a network (or multiple users on one PC), then you may prefer to enter the mailbox details in the Mailtraq Console, where all the options (and on-line documentation) are available.

Collect Mail from a Remote Mailbox

If your service provider supplies e-mail only through a POP3 account, then you can use this facility to collect it.



The 'Leave mail on the server' option allows you to download a copy of the messages, leaving them on the server for another system to download later. Mailtraq will not download a second copy of the messages (providing the server supports this).

- You should check with your service provider to find out if their mail server supports this option.

Provide News Services

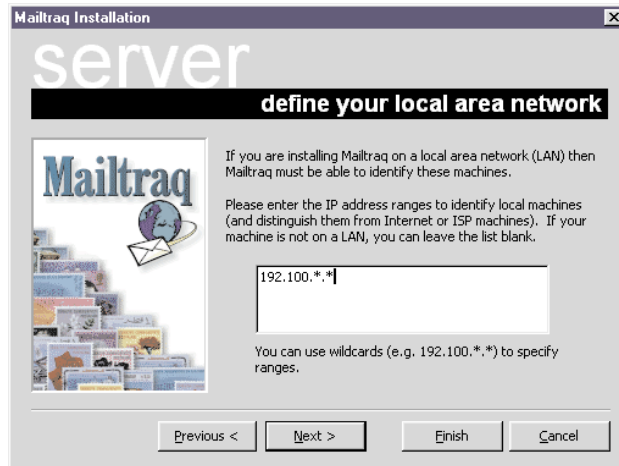
If you wish to collect news articles from your service provider (or any other public news provider), then you can enter the details here. You will only be able to subscribe to news groups after Mailtraq has connected to the news server and downloaded a list of the groups it carries.

Define your Local Area Network

In this section you can define the IP addresses that represent your local area network. These are the IP addresses assigned to the network adapters that connect the machines in your LAN, which should not be

confused with the IP address used by your dial-up adapter (e.g. modem) when connecting to a service provider.

Mailtraq uses these settings to identify the source of a network connection, and this can be used as part of a network firewall.



Do not confuse this local area network IP address with that of your Internet Domain

You do not need to enter the IP address of every machine on your network, as you can replace each of the IP quads with an '*' to represent any number. For example, if all your machines use the IP range from 192.168.0.1 to 192.168.0.255, then you can enter 192.168.0.* to represent all of them.

Provide a Web Proxy

If you wish to have Mailtraq provide one of the web services listed, then you should check this option. If you don't have any other web services running on the host machine, then port 80 is usually the most appropriate.



This is the default web server port, and if you use a web browser and refer to the host machine, web pages will be requested from the Mailtraq Web Server.

If you wish to use a different web server (other than Mailtraq), but still wish to use the web services that Mailtraq provides, then you should select a different port. Usually 8080 is appropriate for a web proxy, although essentially any port can be chosen (as long as each web client is configured appropriately).

Start Automatically

Unless you are experimenting with Mailtraq, you will probably want to have it load automatically in some way.



Mailtraq provides two options :—

■ Start when a user logs in

In a Windows NT system, or a networked Windows '95 machine, you may have to 'log in' (enter a user name and password) before being able to use Windows. Other systems may not need a user to log in at all, in which Mailtraq starts when Windows is loaded.

■ Start when Windows is loaded

With this option, Mailtraq is loaded as a system service as soon as Windows is started. For a multi-user environment, where it is possible that nobody will be logged in to the host machine, this is the most appropriate option.

This facility is handled differently by Windows '95 and Windows NT. Under Windows NT, Mailtraq will appear as a service in the Services applet (accessed from the Control Panel).

3

Quick Start Guide

This chapter provides a deals with a number of common tasks that you are likely to want to perform early using Mailtraq. The step-by-step tutorials will help you configure users and mail slots. This chapter covers :—

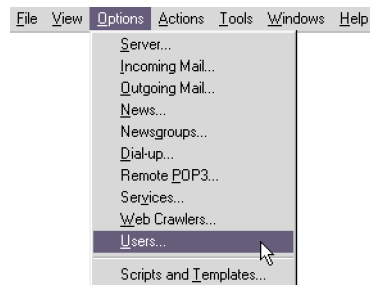
- Creating User Accounts
- Creating Mailboxes
- Creating Mailing Lists and adding Subscribers
- Creating Archives

Multi-User Networks

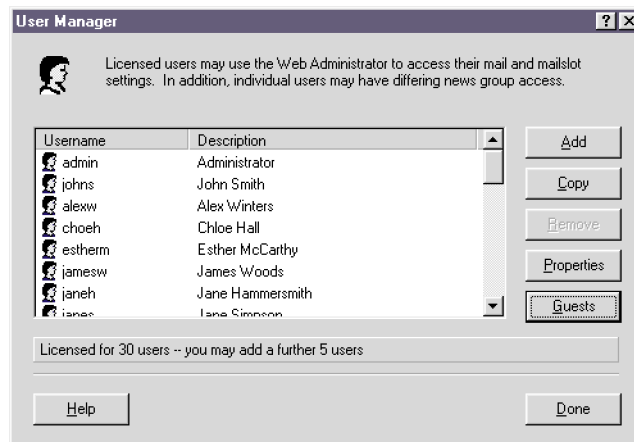
If you are installing Mailtraq in a single-user environment then you can skip this section. When configuring mail slots you should leave the *Owner* property set to the Administrator.

- You can only configure a multi-user system if you have purchased a multi-user Mailtraq license.

To prepare Mailtraq for a multi-user environment you will need to configure the user accounts. This is done in the *User Manager* (accessed from the *Options* menu in the Mailtraq Console).



When you click on the *Users* option, the dialogue below will appear. This is the *User Manager* and user accounts can be created and deleted from here.

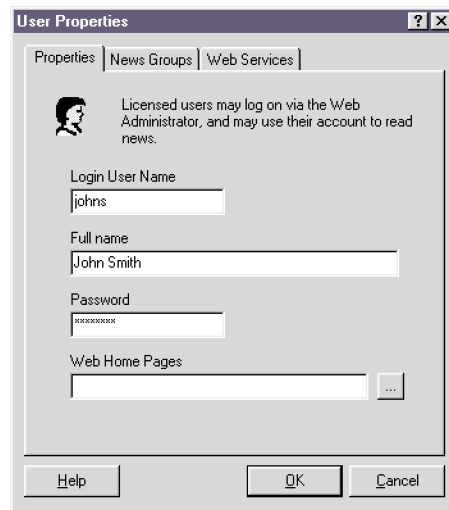


Creating a User Account

To create a new user account, follow these steps :—

- 1 Click the *Add* button (which will create a new user and open the *User Properties* dialogue)
- 2 Enter a short user identifier in the *Login User Name* edit box. This name should not contain any spaces or any characters other than letters and numbers. This is the name that will be used when logging in to the Web Administrator, and will appear in the Console.
- 3 Enter the full name of the user in the *Full Name* edit box.
- 4 Enter a password for this user. The characters entered will be shown as asterisks. Note that the password is case-sensitive.

An example completed
User Properties
dialogue



The user account is now configured, and you can click on *OK* to store the new account. If you click on *Cancel*, the user will not be created, and any changes will be discarded.

You can configure the other properties of the user account at any time by selecting the user account in the *User Manager* and clicking on *Properties*. These properties are discussed later in this manual.

Adding More User Accounts

You should add all the users you wish to configure in the same way as described above. Use the *Copy* button to create a duplicate of the currently selected user account. Since the user details should be unique, this is only useful if you wish to set the other user account properties (such as the news group access).

Creating Mail Slots

In Mailtraq, a *Mail Slot* is a 'recipient' of e-mail messages. That is, if a mail slot called 'johns' is created, then mail sent to 'johns@yourdomain.com' will be handled by that Mail Slot.

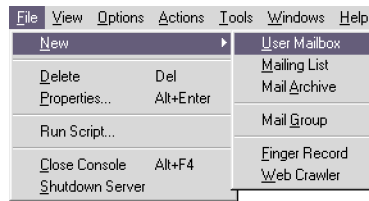
Mailtraq offers three types of mail slot, in addition to three special mail slots (the *postmaster*, *mailserver* and *newsserver* — described later). The three types are *Mailboxes*, *Mailing Lists* and *Archives*.

Mailboxes are folders in which e-mail messages are stored for users to collect with their mail clients.

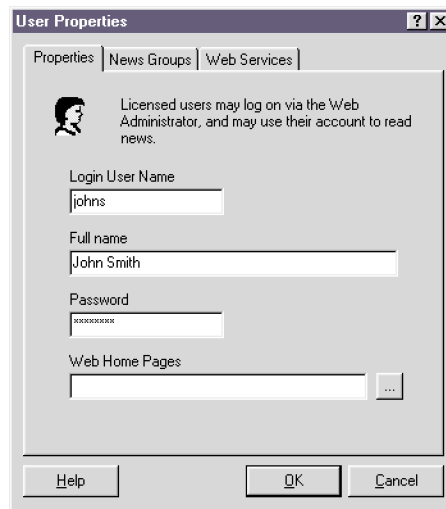
Creating a Mailbox

Follow these steps to create a new mailbox :—

- 1 Click on the *File* menu, and select *New* ➤ *User Mailbox*



The *Mailbox Properties* dialogue will appear



- 2 In the *Mailbox Name* edit box, enter the identifier (which would appear before the '@' symbol in the e-mail address). This must not include any spaces or quotation marks. Unless you have specific reason to, don't use any characters other than alphabetic letters and numbers.

- 3 In the *Name/Description* field, enter the descriptive name of the mailbox. If the mailbox will be the main mailbox for a user, this should be the user's name.
- 4 Enter a POP3 password. This will be used when the mailbox owner collects messages from their mail client. Note the password is case sensitive and the characters entered will be displayed as asterisks.
- 5 Select the owner of the mailbox from the drop down list. This will be a user previously configured in the *User Manager*. If you are configuring a single-user system, you do not need to change this property.

The mailbox is now configured, and you can click on *OK* to save the changes. If you click on *Cancel*, the mailbox will not be created and any settings will be discarded.

You can configure any of the other mailbox properties at any time by clicking on the mailbox in the Console and selecting *Properties* from the *File* menu.

Creating a Mailing List

Mailing Lists simply forward the messages sent to them to a number of other recipients. Mailing Lists are described in detail in Chapter 11. This example will show how to create a basic ‘Distribution’ style list.

- 1 Click on the *File* menu, and select *New* ➤ *Mailing List*

The *Mailing List Properties* dialogue will appear



- 2 In the *List Name* edit box, enter the identifier for the mailing list (the name that appears before the ‘@’ symbol in the e-mail address). The name must not contain spaces or quotation marks. You should use only letters and numbers if possible.
- 3 Enter the full name of the list in the *Description* edit box.
- 4 In the *Maintainer* property, enter the name of a mailbox where you want any administrative messages sent. When, for example, mail cannot be delivered to one of the list members, a message will be sent to this mailbox.

- 5 In the *Owner* property, select the user who will be responsible for the Mailing List. In a single-user environment, you do not need to alter this setting.

Adding Subscribers

Subscribers are the members of the Mailing List (who can either post to the list, or receive messages distributed by the list).

- 1 Select the *Subscribers* Tab
- 2 Click on the *New* button to add a subscriber

The *Subscriber Properties* dialogue will appear

Subscriber Properties

Mailing List Subscriber | Statistics | Filter Messages

Subscribers can be given permissions individually in the mailing list. When subscribers are added via mail, they can receive messages by default.

Full Name: John Smith

Email Address: johns

Send-To Address: johns

Details:

Password:

Flags:

- ☒ Posts Messages
- ☒ Receives Messages
- ☐ Receives Digests (multipart/mixed)
- ☐ Receives Digests (multipart/digest)

Help OK Cancel

- 3 Enter the subscriber's name in the *Full Name* field
- 4 In the *Email Address*, enter the address of the subscriber. This is their 'sending' address and should match the *From:* field of the messages

they post. Mailtraq will use this field to identify the subscriber when they post a message to the Mailing List.

- 5 In the *Send-To Address*, enter the address you want messages sent for this subscriber. Normally, this will be the same as the *Email Address*. However, some subscribers may wish to have mail distributed by the list sent to a different mailbox.

You have now created a subscription for the Mailing List. Click *OK* to keep the new subscriber, or click *Cancel* to erase it and discard the settings.

By default, messages sent to this mailing list will now be sent to this subscriber, in addition to any other subscribers configured.

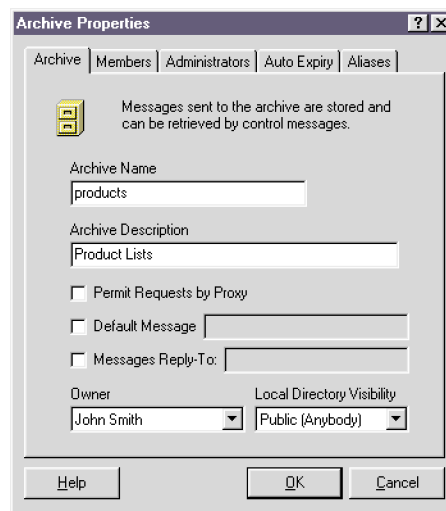
You may modify the subscriber settings at any time by selecting it from the Mailing List *Subscribers* tab and clicking on *Properties*.

Creating an Archive

Archives are similar to mailboxes, but instead of accessing the messages it contains from a mail client, users can send e-mail ‘requests’ to retrieve them.

- 1 Click on the *File* menu and select *New → Mail Archive*

The Archive Properties dialogue will appear



- 2 In the *Archive Name* edit box, enter the identifier of the archive. This is the name that appears before the ‘@’ symbol in the e-mail address. The identifier cannot contain any spaces or quotation marks, and should consist only of letters and numbers (if possible).
- 3 Enter the full name of the Archive in the *Archive Description* field. This can be any short description of the archive (for example: ‘Press Releases’).

- 4 In the *Owner* property, select the user who will be responsible for administrating the archive. In a single-user environment, you can leave this property unchanged.

You have now completed the basic configuration of the Archive, and you can save the settings by clicking on the *OK* button. To discard the settings, click on the *Cancel* button.

You can modify the Archive properties at any time by clicking on the Archive in the Console and selecting *Properties* from the *File* menu.

Retrieving Messages from an Archive

To obtain a list of messages stored in the archive, send an e-mail to *mailserver@yourdomain.com* (where *yourdomain.com* is the domain name for your mail system). In the body of the message, add the line

index archive-name

Replace *archive-name* with the name of the newly created Archive.

4

The Console

This chapter introduces the Mailtraq Console, which is the primary interface for Mailtraq and provides access to all of Mailtraq's facilities.

The chapter covers the following topics :—

- Accessing the Console
- An overview of the Console
- The Console Objects

The console allows the user to configure and manage Mailtraq, as well as monitor it's operations and data. One of the key features of Mailtraq is the way the console gives a real-time view of exactly what is happening. You can view messages in mailboxes, e-mail being delivered, messages

being retrieved from POP3 accounts, and even news articles arriving in news groups.

The console operates completely independently of the other Mailtraq services. This means that you can view mailboxes, read news articles, etc. without interfering with the tasks being carried out simultaneously by Mailtraq.

Accessing the Console

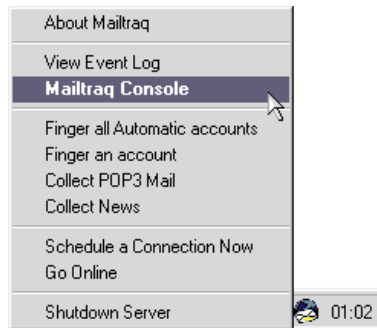
You can open the console at any time while Mailtraq is running. There are three ways to do this :—

- The System Tray Icon. You can double-click on the icon or right-click and select 'Mailtraq Console'.
- The Mailtraq Console loader. By default this is installed in the *Start Menu* within the Mailtraq group, and is given the keyboard short-cut of *CTRL-ALT-M*.
- The Mailtraq Console Applet. You can run the Mailtraq Console applet, which appears in the Windows Control Panel.

The System Tray Icon

When Mailtraq is running an icon will appear in the System Tray on the Task Bar. Occasionally, when Mailtraq is running as a system service, an icon will not appear even though Mailtraq has been started. This is because the application that provides the system tray, Windows Explorer, is a normal application and is started and stopped as users log in

and log off. Mailtraq will try to detect this, but when the icon does not appear you can simply use the Mailtraq Console applet in the Control Panel.

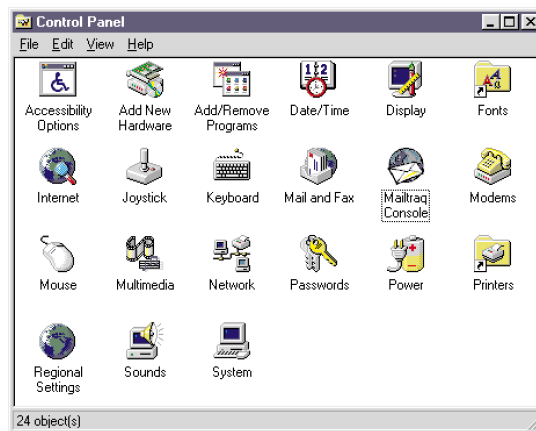


It is also possible to intentionally disable the icon, and this can be done from the *Server Properties* (from *Options* menu in the Console).

The Mailtraq Console Applet

While Mailtraq is running, you can open the Console through the applet in the Control Panel. This may be useful if, for example, you have chosen to hide the system tray icon.

The Windows Control Panel, showing the Mailtraq Console applet

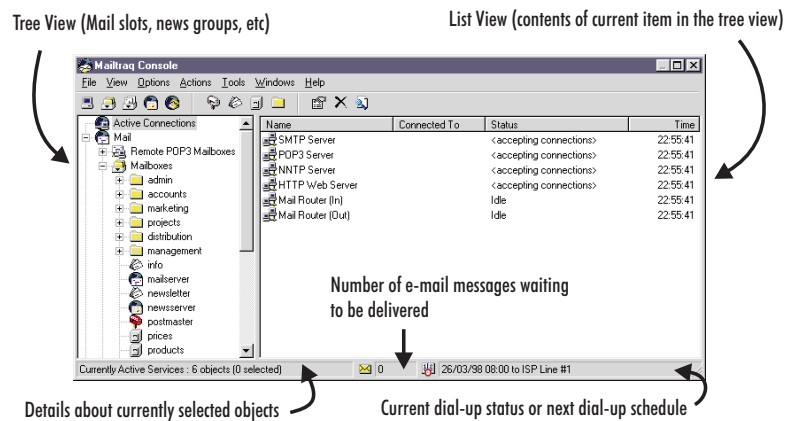


You can prevent unauthorised access to the console by password-protecting it. This can be configured from the 'Administration' tab of the *Server Properties* (from the *Options* menu in the Mailtraq Console). Whether you access the Console from the Control Panel applet, or from the System Tray Icon, the password will still be required to open the Console.

Overview of the Console

The Console uses the standard dual-window design common to Windows Explorer. A hierarchical tree appears on the left, while the objects in the currently selected tree node appear on the right. You can resize either window pane by dragging the vertical bar separating them.

The structure of the Mailtraq Console



The above illustration shows a typical view of the console. In the left window pane (the Tree View), the currently selected object is the 'Active

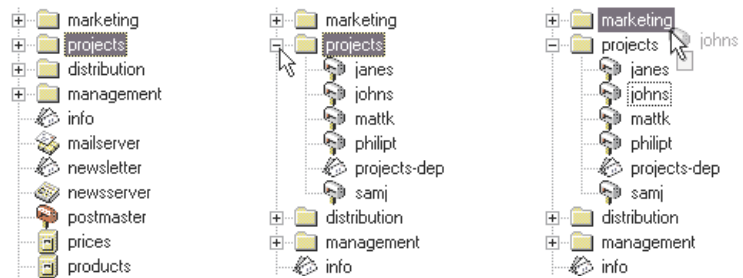
Connections'. In the right window pane (the List View), the current contents of that object are displayed.

The Tree View

Each object in the Tree View represents some sort of container. When you select the object (by clicking on it with the mouse, or by using the keyboard) the contents will appear in the List View on the right.

If the contents are also containers, then they will also appear in the tree as branches extending from the icon in the Tree View.

Navigating the Tree View, and dragging Mail Slots between Mail Groups



The Tree View is represented by a hierarchy of groups showing the relationship between objects

Groups can be opened or closed to reveal the objects they contain

Some objects can be dragged to other groups to change ownership

Most objects have some options associated with them, which can be reached either by the *File* menu (when the object is selected) or by right-clicking on the object.

Mail slots can be moved between mail groups (shown as folders) simply by dragging them (as illustrated above). Mail Groups can also be moved in this manner.

The List View

This view shows the contents of the currently selected object in the Tree View. An example of this is where the current object is a mailbox. In this case the List View will show the messages contained in the mailbox.

The illustration below shows a mailbox, and how the familiar explorer marquee selection method can be used to select multiple messages.

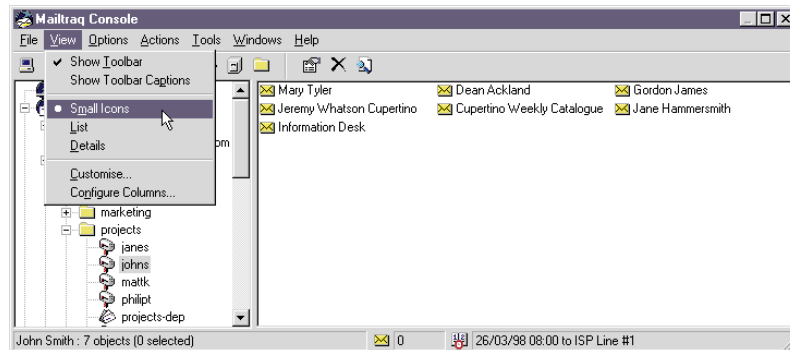
*Selecting objects using
the Marquee method*

From	Subject	Date/Time	Size
✉ Mary Tyler	Fw: Document Review	Sat, Jan 24	24264
✉ Dean Ackland	Fw: Document Review	Sat, Jan 24	1201
✉ Gordon James	Grrt's Courier Service to be changed	Sat, Jan 24	9133
✉ Jeremy Whatson Cupertino	Courier Service Details	Sat, Jan 24	20462
✉ Cupertino Weekly Catalogue	Catalogue Number 55	Sat, Jan 24	27636
✉ Jane Hammersmith	Christmas Party	Sat, Jan 24	579

The illustration above shows the 'Details' view style. You can change this for each type of object from the *View* menu.

The example below shows the same mailbox displayed in the 'Small Icons' style. In any style other than 'Details', only the caption beside the icon is shown. You can change this with the 'Configure Columns' item in the *View* menu.

*Changing the List View
style*



You can also rearrange the order of the columns (in the 'Details' view) simply by dragging the column headings, as illustrated below.

Changing the column
order by dragging the
column headings

From	Date/Time	Subject	Date/Time	Size
✉ Mary Tyler		Fw: Document Review	Sat, Jan 24	24264
✉ Dean Ackland		Fw: Document Review	Sat, Jan 24	1201
✉ Gordon James		Gratt's Courier Service to be changed	Sat, Jan 24	9133
✉ Jeremy Watson Cupertino		Courier Service Details	Sat, Jan 24	20462
✉ Cupertino Weekly Catalogue		Catalogue Number 55	Sat, Jan 24	27636
✉ Jane Hammersmith		Christmas Party	Sat, Jan 24	579

Changing the column order in this way does not affect which column will be used as the caption in the other list views — that must be done with the 'Configure Columns' option.

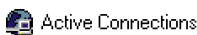
As with the Tree View, you can access a context menu by right-clicking on an object in the List View. You can also access these same commands from the *File* Menu when the object is currently selected.

- *Note:* The date column for a message shows when it arrived in the mailbox, not when it was sent.

The Console Objects

This section describes some of the different types of objects you will find in the console. Each object type is represented by a different icon (graphic), and you can access the context-menu for that object by right-clicking on it's icon.

Active Connections



Active Connections

The Active Connections object is used to show all the current communications tasks and services. Both services (to which other clients connect) and clients (which are connected to other services) are shown here.

Another item which is shown here are monitors which watch other objects (such as disk directories) for activity. The *Inbound* and *Outbound* routers are examples, as are the mail gateways.

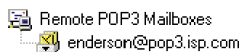
*The Active
Connections list*

Name	Connected To	Status	Time
SMTP Server		<accepting connections>	17:19:07
POP3 Server		<accepting connections>	17:19:07
NNTP Server		<accepting connections>	17:19:07
HTTP Web Server		<accepting connections>	17:19:07
Mail Router (In)		Idle	17:19:07
Mail Router (Out)		Idle	17:19:07
SMTP Client	mail.isp.co.uk	Sending ENDR63300E6 (50%)	17:33:26
SMTP Client	mail.isp.co.uk	Sending ENDR63300E8 (43%)	17:33:26
SMTP Client	mail.isp.co.uk	Sending ENDR63302B2 (4%)	17:33:28
SMTP Client	mail.isp.co.uk	Sending ENDR63300EC (39%)	17:33:30
NNTP Client	news.isp.co.uk	news.isp.co.uk (Downloading 271 of 2717 from comp.lang.pasc...	17:33:32
NNTP Client	news.isp.co.uk	news.isp.co.uk (Downloading 244 of 764 from comp.os.ms-wind...	17:33:32
SMTP Client	mail.isp.co.uk	Sending ENDR63302D1 (6%)	17:33:32
POP3 Client		Connecting to enderson@pop3.isp.com	17:33:55

The illustration above shows the *Active Connections* window with a number of tasks in progress. You can cancel any of these tasks by right-clicking on them and selecting the 'Close' option.

- *Note:* Mailtraq will usually continue the task until it is safe to suspend. Sometimes the task cannot be closed while Mailtraq is waiting for a network response. For these reasons, you should not expect an immediate reaction from Mailtraq.

Remote POP3 Mailboxes



The Remote POP3 Mailboxes represent accounts that have been configured with other POP3 Mail Servers. Mail can be downloaded from each account and placed in mail slots in Mailtraq, or processed in other ways (as with mail arriving from any other source). Refer to page 114 for details on the remote POP3 collection facilities.

Mail Groups



These objects are used as containers for mail slots. They represent logical groupings rather than any formal ownership, and a single user's mail slots may appear in several different mail groups.

With systems that have large numbers of mail slots, mail groups can be an effective way to organise them.

Another important use of mail groups is mail slot accounting. Settings such as the maximum amount of mail that can be stored in a mailbox, are applied to the mail group (rather than mail slots individually). Mail slots in that group are then subject to those settings.

Mail Slots



Mail Slots are the mail receivers in Mailtraq. Mail handled by Mailtraq must either end up in mail slots, or be forwarded to another mail server.

There are three basic types of mail slot :—



Mailboxes



Mailing Lists



Public Archives

■ Mailboxes

These are used to store mail for users. Mail can be accessed from these objects via POP3 (using mail client software) and using the Web Administrator

■ Mailing Lists

These mail slots distribute mail to their subscribers. In fact mailing lists are very powerful and perform many tasks other than simply forwarding mail. They are described in detail in Chapter 11. Mail that is waiting to be distributed in the form of a digest will be stored in the mailing list until delivery time.

■ Public Archives

These mail slots store messages for public access. They are often used in association with Mailing Lists (and are also discussed in Chapter 11). The messages contained in the archive can be viewed with the Web Administrator, but are most often retrieved by sending special messages requesting them.

Other than these three types of mail slot, there are also three special permanent slots :—



■ Postmaster

This is similar to an ordinary mailbox, but it is where Mailtraq will send various reports and notices, and it a requirement of the Internet that mail addressed to the 'postmaster' be accepted and read by an appropriate administrator.

■ Mailserver

This object simply processes commands within messages. For example, to join a mailing list called 'newsletter', you would send a message to 'mailserver' with the line

```
subscribe newsletter
```

in the message body. The *mailserver* object is discussed in more detail with regards to mailing lists on page 130.

■ Newsserver

This object simply takes an e-mail message and passes it to the news service. Refer to page 92 for more information.

Outgoing Mail



Outgoing Mail

This object stores the messages waiting to be delivered to other mail servers. Messages wait here until a connection is made to the appropriate mail server, and then they are sent and finally removed. Messages are not erased until they have been safely transferred to some other location (whether to another mail server, or returned to the sender because of problems).

Before a message can be sent, Mailtraq must decide where to send it. How Mailtraq determines this will depend on the routing tables. Sometimes Mailtraq will not be able to determine where to send a message until it has a dial-up connection and it can look up a 'Mail Exchange' from a Domain Name Server (DNS).

To show the progress of each message, a different icon is displayed at each stage (as illustrated below) :—

*The stages of outbound
mail routing*



Once a message has had a route assigned to it, even if you change the routing tables the messages may not be updated. You can clear the routing assignments with the ‘Clear Host Assignments’ command from the Outgoing Mail context menu.



Subscriptions



The News Subscriptions contain both local news groups, and subscriptions to news groups on remote news servers. One level below the subscriptions object are the news servers.

These objects refer to the servers from which Mailtraq will collect news. The first news server will always be ‘Local Groups’, which contains the groups that are maintained only on this server.

When a news group is selected, the List View will display the news articles currently stored by Mailtraq. The information displayed is similar to mailboxes, as news articles are very similar to e-mail messages.

From	Subject	Date/Time	Size
Manfred ...	What's new? 7,400+ links on Object-Orientation ...	17:33:36	4315
Digispec	Re: Printer port!	17:33:36	1619
Yu Hai an	Re: Help! LIBC.LIB libaray not found with nmak...	17:33:37	11289
Ken	Modify resources in an executable file.	17:33:37	1260
Carl Maill...	Re: Modify resources in an executable file.	17:33:38	1435
Frank	Re: Modify resources in an executable file.	17:33:38	1576

Outgoing News



The Outgoing News object stores news articles that have been posted to news groups subscribed to by Mailtraq. These will be articles that have been posted locally, and they are stored here while waiting to be uploaded to the appropriate news server.

Web Crawlers



The Web Crawlers are used to automatically download web pages and files and insert them into the global web cache. This makes it possible to browse the pages (or even entire sites) completely off-line through the Web Proxy service.

The crawlers can automatically update the pages, and index them for searching (through the Web Administrator).

You can initiate a web crawler from the 'Crawl Now' option on the context menu for the crawler objects.

Users



The Users represent the people who can use Mailtraq to maintain their mail slots through the Web Administrator, access the web through the Web Proxy and read news groups.

For each user in the Console a complete list of all their mail slots are shown in the List View. You can change the ownership of mail slots by dragging them between different users.

Finger Records



The Finger Records are used to retrieve information from various servers. For example, some service providers will offer a 'Finger' service where users can get access to information about their mail store (such as how much mail is waiting).

The finger responses are plain text and can be read by selecting 'View Record' from the context menu of the finger objects, as shown below.



5 Internet Dial-up

This chapter covers Mailtraq's Dial-Up facilities, which allow it to manage access to the Internet. The following topics are included :—

- Configuring the Internet Connection
- Dial-Up Networking
- Manual Dial-Up
- Scheduled Dial-Up

Most installations of Mailtraq will be using the Internet for some services, but without a permanent Internet connection. Mailtraq provides a number of methods to connect to the Internet, and supports connection via third-party services (such as dial-up routers).

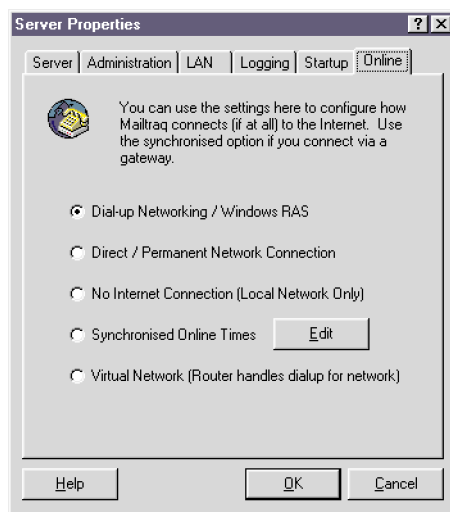
It is important to configure Mailtraq to use the Internet connection effectively. For example, if you have a network router that connects to your Internet Service Provider whenever there is Internet network traffic, then from Mailtraq's point of view you have a permanent connection. However, with a permanent connection Mailtraq will simply initiate outbound mail delivery whenever mail is ready to send — which may result in unnecessarily frequent dial-up connections. To help this, Mailtraq can be configured to only use the router to connect under certain conditions, or at certain times.

Configuring the Internet Connection Type

How and when Mailtraq uses Internet connections can be configured in the Server Properties (available from the *Options* menu in the Mailtraq Console).

The illustration below shows the available options :—

Selecting the Internet connection type from the Server Properties



Dial-Up Networking / Windows RAS

When this option is selected, Mailtraq will initiate a DUN connection whenever it needs to connect to the Internet. It can be configured (elsewhere) to connect at specified times, and under specified conditions, but it will only consider itself 'online' when a DUN connection is active. Most of these settings are configured in the *Dial-up* properties (available from the *Options* menu in the Mailtraq Console).

Direct / Permanent Network Connection

This option should be used when your network is connected to a service provider via a leased line, or when you have a network router and are not concerned about the dial-up frequency. Mailtraq will always consider itself 'online', and will (for example) deliver outbound mail as soon as it is ready.

No Internet Connection (Local Network Only)

With this option selected, Mailtraq will never consider itself 'online', and will never attempt to go online through a Dial-up connection (if one

exists). Use this option if you only wish to provide services within your local area network, or provide a local Intranet.

Synchronised Online Times

This option allows Mailtraq to consider itself 'online' at predetermined times. Use this option when another system handles Internet connections at specific times, or if you limit a dial-up network router.

*Configuring Mailtraq to
synchronise it's on-line
times with another
system*

Synchronised Online Times

Use this form to indicate what times Mailtraq should assume that it has a network connection to the Internet. Click on a time block to mark it as online.

	SUN	MON	TUE	WED	THU	FRI	SAT
0:00							
1:00							
2:00	■	■	■	■	■	■	■
3:00							
4:00							
5:00							
6:00							
7:00							
8:00		■	■	■	■	■	■
9:00		■	■	■	■	■	■
10:00		■	■	■	■	■	■
11:00		■	■	■	■	■	■
12:00		■	■	■	■	■	■
13:00		■	■	■	■	■	■
14:00		■	■	■	■	■	■
15:00		■	■	■	■	■	■
16:00		■	■	■	■	■	■
17:00		■	■	■	■	■	■
18:00							
19:00							
20:00							
21:00							
22:00							
23:00							

Offline
Online

Each block represents 15 minutes.

Cancel Done

Click on the grid to toggle a time block

You can specify the times by clicking on the 'Edit' button and highlighting the fifteen minute periods when Mailtraq can use the Internet services.

In the above example, the network is available from 8:30am to 5:30pm Monday to Friday, and for fifteen minutes every day at 2:00am.

Virtual Network

This option is very similar to the Dial-up Networking option, but Mailtraq will not actually initiate the connection. Scheduled dial-ups can still be configured, allowing Mailtraq to consider itself 'online' as though it initiated the connection, ending the session after an idle time-out.

If you have a dial-up network router (which handles connections to the Internet as needed) which you do not wish to use unnecessarily, then this can be an effective solution. You can use the Console 'Go Online' and 'Go Offline' to manually change Mailtraq's state with this option.

- *Note* that with any of these options, if Mailtraq goes offline while services are active, it will not actually disconnect them unless the network connection is broken. For example, if the 'Synchronise' option is selected, and a large message is being downloaded, then Mailtraq will not terminate it before completion.

Dial-Up Networking

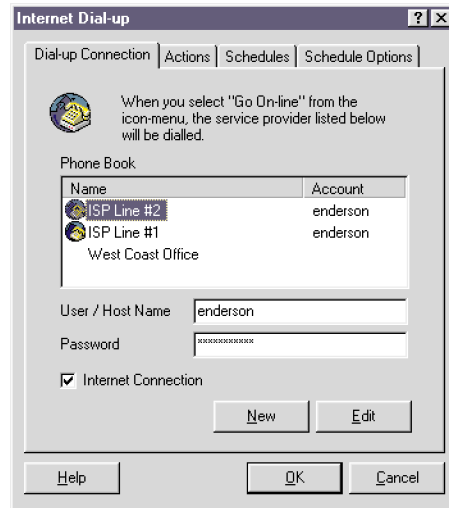
Mailtraq uses Internet services, and communicates with other machines on the Internet, through the TCP/IP networking layer in the operating system on which it is installed. If you connect your network to the Internet via a modem (analogue or ISDN) then the connection is handled by Dial-Up Networking (or the Remote Access Service).

The configuration of the Dial-Up Networking connections will depend largely on your Internet Service Provider, and the nature of your network. Once you have configured one or more Dial-Up Networking (DUN) connections, Mailtraq can be configured to use them.

You can select which DUN connections Mailtraq can use (and the username/passwords needed to initiate them) from the *Dial-up* entry from the *Options* menu in the Mailtraq Console.

The illustration below shows two connections identified for use by Mailtraq.

*Selecting Dial-Up
Networking connections
that Mailtraq can use
for on-line services*



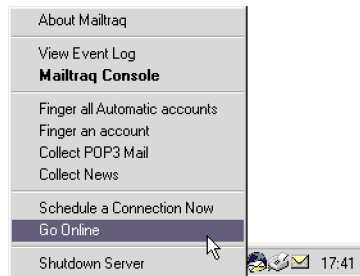
For each connection that you wish Mailtraq to initiate, you must provide the username and passwords needed when logging on to your Internet Service Provider (even if you have already entered this information elsewhere in Windows).

Mailtraq will only consider itself 'on-line' when a connection that has been checked as 'Internet Connection' becomes active. Mailtraq constantly monitors these connections, so it can use them even if another program initiates them.

Manual Dial-Up

On the *Actions* menu of the Console, and on the *System Tray* icon menu, you will find the 'Go Online' option. This will initiate the currently selected connection.

Manually initiating a dial-up

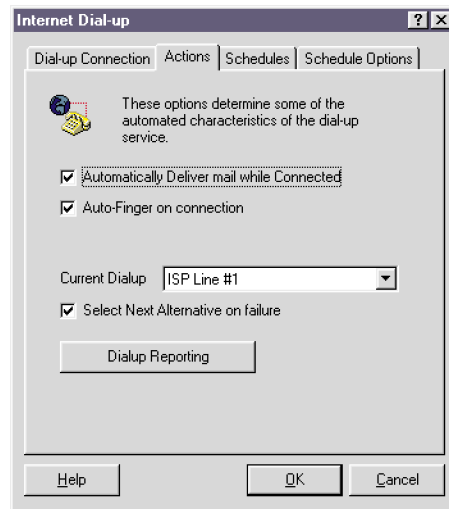


A manual connection will not be automatically disconnected by Mailtraq, but will remain active until you disconnect either via the 'Go Offline' command, or by closing the DUN connection elsewhere in Windows.

- *Note* that some service providers will automatically close the connection after a period of inactivity as a safety precaution. Windows can also be configured to do this.

The actions that take place when you use the *Go Online* command are determined by the settings in the 'Actions' tab of the Dial-Up Properties dialogue, as shown below.

*The Actions tab of the
Dial-Up Properties*

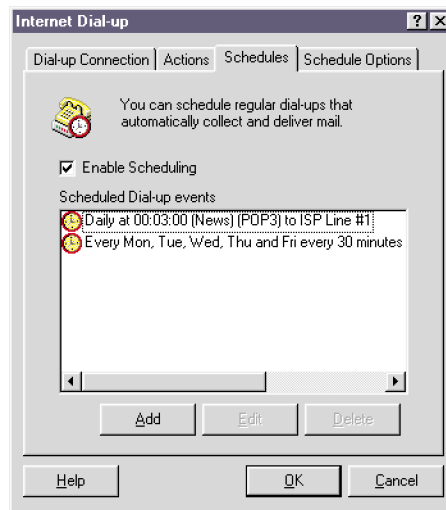


The 'Select Next Alternative on Failure' option allows Mailtraq to automatically change the Current Dialup to the next available option when a dialup cannot be completed. If your service provider has more than one telephone line, you can configure a separate DUN connection for each, and automatically switch to an alternative using this method.

Scheduled Dial-Up

Mailtraq provides a comprehensive scheduling facility to manage automatic dial-up connections (and disconnections). You can configure these from the 'Schedules' tab of the Internet Dial-up properties dialogue, as illustrated below.

*Configuring Dial-Up
Schedules*

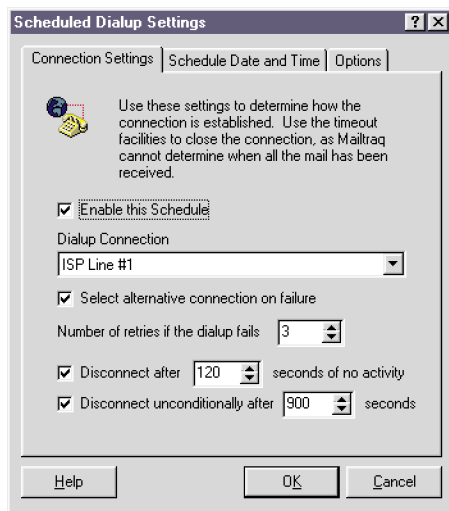


This tab shows all the schedules currently configured, identified by a brief summary (which does not include all the schedule options). The icon beside each schedule will appear grey if the schedule is disabled.

Configuring Schedules

Configuring a Scheduled Dial-Up

When you create or edit a schedule, you will see the Schedule Properties dialogue, shown below :—



Select alternative connection on failure

This option allows Mailtraq to automatically change the selected Dialup Connection when a dialup cannot be established. If your service provider supplies multiple telephone numbers, then this can be an effective way to automatically deal with busy lines or line problems.

Number of retries if the dialup fails

This setting allows Mailtraq to automatically start the connection again when it cannot be successfully established. These attempts will only take place if the Dial-Up Networking system reports the connection as having failed. Occasionally a connection will appear to be successful, but will not actually allow network traffic (because of a problem with the service provider's equipment, for example). Mailtraq cannot detect this, and will consider the connection successful, even though it will fail to perform its scheduled tasks.

If delivering outbound mail is essential, you can configure schedules that act only if mail is waiting, and configure them to take place frequently.

Disconnect after...

Mailtraq closes the connection automatically when it detects that it is no longer in use, or after it has been active for a specified period of time. Since many of the services that use the connection are passive (i.e. initiated by other machines), Mailtraq cannot determine exactly when the connection is no longer required. Instead, it monitors its own activity on the connection, and when a period has passed when no activity has taken place, it will close the connection.

It is very important to set an unconditional time-out to prevent a connection from being kept open unnecessarily.

- *Note* that Mailtraq can only detect activity that it is responsible for. It cannot detect other applications using the connection (such as web browsers or FTP programs) unless they are using Mailtraq as a proxy.

If you have the 'Query Connect/Disconnect' option enabled (from the Schedule Options tab on the Dial-Up properties dialogue), then a window will appear shortly before the connection is automatically disconnected. From this window you can have the automatic disconnection canceled, or temporarily suspended.

*Preventing the
schedule's automatic
disconnection from
occurring*

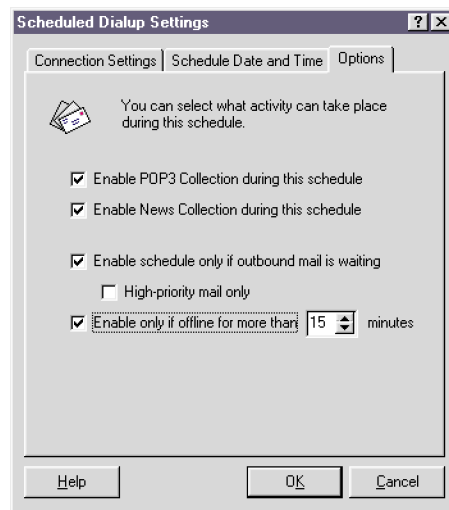


If you cancel the disconnection, either from this window or from the 'Cancel Automatic Disconnection' command, then the connection will become manual and will no longer be subject to the schedule options (such as whether or not POP3 collection is enabled).

Schedule Options

A number of options are available for the scheduled dial-ups, which can be configured in the *Options* tab of the Schedule Settings dialogue (shown below).

The Scheduled Dial-Up Options



Enable POP3/News Collection during this schedule

If this option is not checked, then automatic POP3 or News collection will not take place when the schedule is active. They can still be initiated manually from the *Actions* menu in the Console.

Enable schedule only if outbound mail is waiting

If this option is checked, then the schedule will be ignored while the *Outbox* is empty. This is similar to the Automatic Schedule initiated if there is outbound mail waiting, but because it is scheduled according to a date and time it is therefore predictable.

Checking the 'High priority mail only' option will enable the schedule only if mail in the Outbox has the High Priority flag (shown as a red stripe through the message icon).

This does not apply to messages that have been delayed due to a delivery problem. Such messages are only considered 'waiting for delivery' when they reach their 'delayed' time (which can be made visible through the *Configure Columns* option in the *View* menu). Delayed messages are identified by a small circular clock on the envelope icon.

**Enable only if offline
for more than...**

If this option is checked, then the schedule will only be enabled if it would take place no less than the specified period since Mailtraq was last 'on-line'.

Dial-Up Schedule Options

These options are available from the fourth tab of the Dial-Up Properties dialogue, which is shown below :—

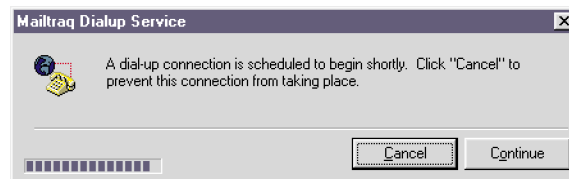
*The Schedule Options
tab of the Dial-Up
Properties dialogue*



**Query
Connect/Disconnect**

If this option is enabled, then a window will appear shortly before a scheduled dial-up takes place asking the user to confirm. An example of such a window is shown below :—

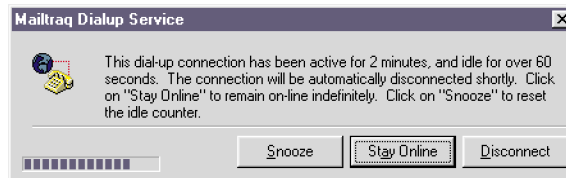
*Offering the user the
option to cancel a
connection*



If neither button is pressed, then the connection will continue automatically when the progress bar has elapsed.

Shortly before the connection is automatically disconnected, another window will appear asking the user to confirm the disconnection. This window also has the option to reset the idle time-out (the Snooze button), but this button will not appear when the unconditional time-out is reached.

The user can be given the option to postpone or cancel a disconnection



If the 'Stay Online' button is pressed, then the connection becomes manual (requiring the user to disconnect manually with the 'Go Offline' command). This also means that the schedule options (such as whether or not POP3 is enabled during the schedule) will no longer have any effect.

Warn on Missed Schedules

If this option is selected, then when Mailtraq is started it will determine if a schedule would have taken place since it was last shut down. If a schedule appears to have been missed, then it will display a window asking the user if they wish to restart the schedule.

Auto-Schedule when outbound mail is waiting

If this option is checked, then when mail is waiting in the outbox the 'Schedule Now' command will be issued (providing a schedule is not already due within the specified time).

This does not apply to messages that have been delayed due to a delivery problem. Such messages are only considered 'waiting for delivery' when they reach their 'delayed' time (which can be made visible through the *Configure Columns* option in the *View* menu). Delayed messages are identified by a small circular clock on the envelope icon.

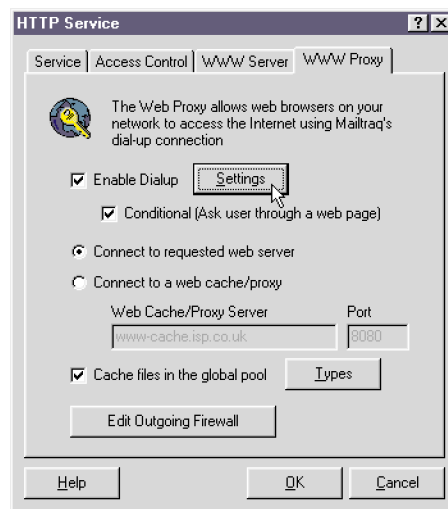
Automatic Schedules

Mailtraq can also initiate a schedule for a specific purpose, such as to deliver mail or allow a web proxy to operate. It is also possible to initiate a schedule manually with the 'Schedule Now' command (from the *Actions* menu in the Console, or from the *System Tray* icon menu).

The advantage of this type of connection is that it will be automatically disconnected when it is no longer needed.

An example of this type of dial-up is the automatic schedule used to connect when Mailtraq needs to proxy a web connection.

Configuring the dial-up
schedule initiated by the
WWW Proxy



When an automatic schedule takes place, the options for that schedule apply while it is active. For example, if the Web Proxy schedule does not permit News collection, then no news collection will take place automatically.

6

Introducing E-Mail

This chapter introduces the concept of e-mail, and how it is used and delivered. The following topics will be covered :—

- What is E-Mail?
- Message Envelopes
- Message Headers
- The Delivery Protocols (SMTP and POP3)
- Mail Routing and Mail Exchanges
- Mail Gateways

This chapter will also cover some distinctions between the different types of delivery protocol that may affect your choice of delivery method.

What is E-Mail?

Electronic mail was one of the first practical uses of wide area networking, and remarkably, little has changed since it began. The content has certainly changed: now you can send graphics, sound and video, as part of a message. No doubt, the future will provide even more creative uses of e-mail. Thankfully, none of these new forms of message affects the actual delivery of them. E-Mail messages have a simple and consistent structure, leaving mail servers to do the task of transferring them without knowing the complexities of what they contain.

An e-mail message consists of a header and a body. The header is normally not visible to the author and recipient of the message, but instead tells the e-mail client software (in which it was written and read) about the message. The body represents what the user can actually see.

*An example e-mail
message (shown with
message header)*

<input type="radio"/>	To: John Smith <john@enderson.com>	<input type="radio"/>
<input type="radio"/>	From: Jane Simpson <janes@enderson.com>	<input type="radio"/>
<input type="radio"/>	Subject: Contact Mr Green	<input type="radio"/>
<input type="radio"/>	Date: Tue, 28 Oct 1997 11:50:58 -0000	<input type="radio"/>
<input type="radio"/>	Mr Green called at 12:30. Please contact	<input type="radio"/>
<input type="radio"/>	him as soon as possible.	<input type="radio"/>
<input type="radio"/>	--	<input type="radio"/>
<input type="radio"/>	Jane Simpson	<input type="radio"/>
<input type="radio"/>		<input type="radio"/>

Amongst the headers of an e-mail you will find the *From:* and *To:* fields. These describe the author and original recipient of the message, and provide both the e-mail address, and usually the name. For example, you may see the header field

To: John Smith <john@enderson.com>

or

To: john@enderson.com (John Smith)

Both of these are quite valid, as are a myriad of other formats. In fact, this information does not actually indicate to whom the message is to be sent. Imagine you wish to write a message to John Smith, but since he is away on vacation, it is passed on to Jane Simpson (his secretary). She will still see that the message was addressed to John, but if that is the case, how did she receive it? Because the actual recipient is described in the message envelope, not the message itself.

Message Envelopes

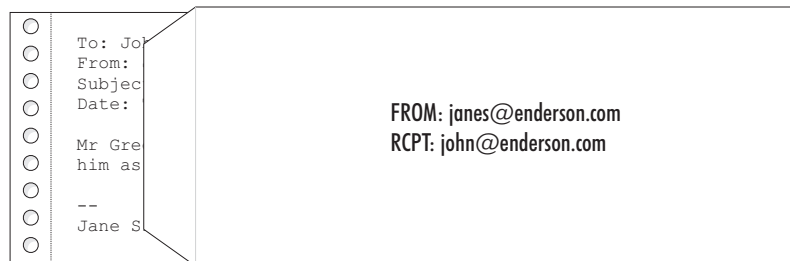
Just as with real letters, the person on the top of the first page does not indicate to whom the message is sent. The message is actually sent to the person on the envelope. If the sorting office at Enderson Enterprises gets the message, they can just put the letter in another envelope for Jane Simpson.

E-Mail works in a similar way. During delivery, each message also has an envelope. This envelope indicates who sent it, and where it is going. The envelope can be altered during delivery without affecting the message itself.

The envelope also contains a return address. This address may not be the person who wrote the letter, but usually the last step in the delivery process (whoever put this message in the envelope). If the message gets stuck somewhere (because it cannot be successfully delivered), the sender

must be informed. Because the contents of the message may vary widely, the delivering agent only has the envelope sender to notify. Thus, during delivery only the information in the envelope is used.

The Message Envelope



Once a message has been delivered (or put in a mailbox for collection), the envelope is usually discarded. This is because the only recipient is the person who comes to collect the mail. Thus, it is useful to think of the envelope as being part of the delivery process, not part of the message.

Message Headers

There are a number of header fields that you can expect to find in any message. The meaning of these fields has been standardised to avoid confusion, and the table below describes the most common headers.

Many mail programs add additional information to the header, but if this information is not part of the standard, the header field is prefixed with an X-.

Header Field	Purpose
To:	The person to whom the message was written (not necessarily the recipient).
From:	The author of this message.
Sender:	The entity who is responsible for sending the message (they may not have actually written it).
CC:	Carbon Copy recipients. These are additional entries similar to the To: field.
Reply-To:	This field indicates where replies to the message should be sent. For example, if the message is a questionnaire the author may have no interest in the responses, and so directs them to the analysis team.
Subject:	A brief (one-line) description of the message.
Date:	The date at which the message was written.
Received:	This is the only field that normally appears more than once in a message. Each instance provides details about one message 'hop' during the delivery process. These are also called the "Trace Fields", as they provide useful details on the message's traveling history.

There are many other fields that you will encounter, but they are not covered in this document. The official reference is RFC822, which you will find further details on in the bibliography.

*An Example Message
showing the Message's
Header Fields*

○	Return-Path: <dan@smartdrive.co.uk>	○
○	Received: from pop3.demon.co.uk by fastrag.co.uk	○
○	with POP3 (Mailtraq/0.8.2.54) id FSTR97196A94;	○
○	Mon, 26 Jan 1998 15:00:57 -0000	○
○	Received: from punt-2.mail.demon.net by mailstore for elric@fastrag.demon.co.uk	○
○	id 885823283:15:05054:1; Mon, 26 Jan 98 14:01:23 GMT	○
○	Received: from post-10.mail.demon.net ([194.217.242.154])	○
○	by punt-2.mail.demon.net id aa0616787; 26 Jan 98 14:01 GMT	○
○	Received: from smartdrive.demon.co.uk ([194.222.222.105]) by post.mail.demon.net	○
○	id aa1025750; 26 Jan 98 14:00 GMT	○
○	Received: from 110oct ([100.100.100.9]) by smartdrive.co.uk	○
○	with SMTP (Mailtraq/0.8.3.56) id SMRT971748D182;	○
○	Mon, 26 Jan 1998 13:57:01 -0000	○
○	From: Dan Lyon <dan@smartdrive.co.uk>	○
○	To: Elric Pedder <elric@fastrag.co.uk>	○
○	Subject: Example Message	○
○	Message-ID: <SMRT971748D184@smartdrive.co.uk>	○
○	Date: Mon, 26 Jan 1998 13:54:24 -0000	○
○	X-MSMail-Priority: Normal	○
○	X-Priority: 3	○
○	X-Mailer: Microsoft Internet Mail 4.70.1161	○
○	MIME-Version: 1.0	○
○	Content-Type: text/plain; charset=ISO-8859-1	○
○	Content-Transfer-Encoding: 7bit	○
○	X-Hops: 2	○
○	This message contains an example of the	
○	headers found in an e-mail. Notice	
○	received headers, one added for	
○	has carried the message	

Simple Mail Transfer Protocol

SMTP is the Internet standard for e-mail transport, and this is how Mailtraq handles mail delivery. While SMTP may be a very simple protocol, it has been proven through many years of use, and it provides a remarkably reliable transport service. The SMTP server listens on a TCP/IP Port. You do not need to know much about TCP/IP to manage Mailtraq, but it is useful to know that servers listen on ports, and clients connect to those ports. All the client needs to know is which machine to connect to, and the port on which the service is available. SMTP, for example, usually listens on port 25. No two services can listen on the same port, but more than one client can connect to that port.

When an SMTP client connects to the server, they start a conversation. During this conversation (called the protocol) they exchange information and the messages themselves. Following is a simplified example of this conversation.

Server / Client Message

Server	Hello, I am the Enderson Enterprise SMTP Server
Client	Hello, I am the Jefferson Associates Message Transport Agent
Server	Okay, we'll accept mail from you
Client	I have a message from James Jefferson
Server	Okay, we'll accept mail from him
Client	This message is for John Smith
Server	Okay, we will accept mail for him
Client	Here's the message
Server	Okay, correctly received and stored
Client	I'm finished
Server	Good-bye

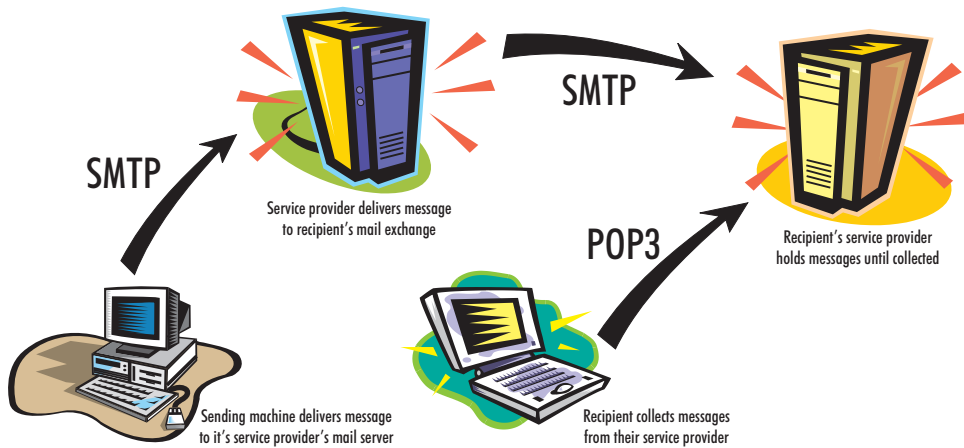
If you actually watch an SMTP transaction, you will see that it is remarkably close to this. SMTP also has many safety features. For example, notice how the server indicates that the message was correctly received and stored. Until the client receives that notice, it will not consider the message delivery successful. Thus, if the server machine fails halfway through receiving the message, the client will know it has to try again some other way (or notify the sender of the message about the problem).

There are also several opportunities for the server to refuse to accept mail. This makes it possible to safely stop a message delivery without having to go through the (possibly long) process of receiving it.

As mentioned earlier, the SMTP Server can accept several connections at the same time. Each connection is handled as a separate, independent task (in Mailtraq, each is handled by a separate thread). Thus, if a

connection is very slow (for example, if the client machine is many thousands of miles away) this will not impede the performance of other channels of communication.

If either the client or the server fails to acknowledge or respond for a long period, the other side will assume there is a fault, and break the connection (either trying again later, or using an alternate route).



Messages are transferred with SMTP, and collected with POP3

Mail Routing and Mail Exchanges

It is one thing to know how to deliver the message, but how does a mail server decide where to deliver it? Each server on the Internet will have a number of rules that help it decide this. To help these servers, they can also look up the Mail Exchange for a given address.

All hosts on the Internet are defined by their unique IP address. Each IP address consists of four numbers (in the range 0 – 255) separated by dots (e.g. 192.100.100.102). These numbers are difficult to remember, so each host is also given a textual name (also called a Domain Name). Servers on the Internet that can translate names to addresses are called Domain Name Servers (DNS). These machines can also provide other types of data for hosts, such as a list of designated Mail Exchanges.

A Mail Exchange (MX) is a mail server configured to accept mail for a given domain, and mail can be sent to these servers using SMTP.

Another way to get mail to its recipient is to pass the message on to a Smart Host. This machine will store the message and then do the MX lookup and subsequent delivery itself. Most Service Providers will supply a Smart Host to simplify your configuration. However, since Mailtraq can use a Smart Host, Mail Exchanges and routing tables, you may decide which methods to choose.

Since Service Providers have to handle messages for a large number of customers, their machines occasionally become overworked. In these cases, Mailtraq can bypass this bottleneck by delivering mail directly to the Mail Exchanges. However, when messages are to be sent to a large number of recipients, it is quite often faster to deliver a single copy of the message to a Smart Host, and leave the bulk of the work to its hands.

The Post Office Protocol

As discussed earlier, the user's mail software will also need a way to obtain the messages from a server. Mailtraq can provide access to the mail it stores through the Post Office Protocol (POP3). Mailtraq is also able to collect mail on behalf of users using this protocol.

Like SMTP, this protocol is used to transfer messages between two machines. However, POP3 is far more versatile (and naturally more complex). Using POP3, a client can examine messages in a mailbox, download a selection of them, and delete them from the mailbox. Because deleting the messages from a mailbox is optional, it is possible to share a mailbox between several users.

Since it is possible to examine a message before deciding to download it, Mailtraq can also use POP3 to help avoid downloading unwanted messages. The examination is simply the opportunity to determine the size of the message, and retrieve only the message headers.

Choosing between SMTP and POP3

If your service provider offers both types of e-mail delivery, you are in a position to select one of them. Unless you configure Mailtraq to specifically collect via POP3, all mail will be received with SMTP. Collection via SMTP requires no configuration: as long as the SMTP service is running, your service provider will attempt to deliver mail to it.

*The *ENV extension,
and the use of
additional header fields
are specific to service
providers — consult
their technical support
service*

If you are considering collecting mail from one POP3 mailbox, and distributing it on your network, then there is a very important issue to consider. Can you rebuild the message envelope correctly? Without the message envelope, it is not possible to determine the message recipients. Mailtraq provides a number of ways to do this, but without help from the POP3 server, they are unlikely to be effective. Mailtraq can obtain the envelope by examining header fields that some service providers add for that purpose (such as the *X-Apparently-To:* field). Mailtraq can also use the *ENV extension to POP3 to obtain the original envelope. If these options are not available, then Mailtraq can examine the *To:* header and extract addresses from that.

Although extracting the recipients from the *To:* header may seem an adequate solution, there are further problems. Consider a message posted to a mailing list. The *To:* header may contain the address *politics@ourworld.net*. It is unlikely that your mailbox received this message because it happened to be addressed as *politics@ourworld.net*, but rather because it is a subscriber to that mailing list. The problem is that if this address was extracted from the *To:* header, the message would be sent back to the mailing list and end up in an endless loop.

To solve this problem, Mailtraq must be configured with a set of rules to ensure that messages to these addresses are correctly handled. Alternatively, all mail downloaded from a POP3 server can be forced into a single mailbox.

If you can overcome these problems, then collection of mail via POP3 may indeed be viable. SMTP is generally regarded as a more robust protocol than POP3. One reason is that if a problem occurs, SMTP tells the client the nature of the problem. POP3 can only indicate success or

failure. There are defined standards for how to handle the different responses that SMTP generates, such as returning a delivery failure report to the sender of the message, or trying an alternative delivery route.

However, once a message is downloaded via SMTP, it is taken out of the mail server's queue and is no longer available. Using POP3 it is possible to analyse the message headers and select only mail addressed to a specific person, leaving the other messages for someone else to collect.

Below is a table comparing the facilities that these two protocols offer :—

Feature	SMTP	POP3
Unwanted or mis-addressed mail can be returned to the sender without downloading	Y	N
Mail can be refused based upon recipient or sender	Y	Y
Mail can be refused based upon other header fields	N	Y
Mail can be refused based upon message size	N ¹	Y
Mail can be left on the server after downloading	N	Y ²

- 1 Mailtraq does provide facilities to defend against oversized messages, by disconnecting the download after the message size has exceeded the set limits, and barring the message next time delivery is attempted.
- 2 Not all POP3 servers support this feature.

Mail Gateways

Another method for delivering mail on Local Area Networks (particularly those without TCP/IP installed) is the use of shared directories (also called gateways). Mailtraq supports a number of gateways, including Pegasus and KA9Q. The idea is that a program writes the message to a specific directory, which is monitored by another program.

It is also possible to write simple, custom e-mail software that can communicate using this method, as it is usually far easier to implement than the TCP/IP solution.

7 Mail Slots

This chapter explains Mailtraq's concept of *Mail Slots*, which are the 'recipients' of e-mail in a messaging system. The Mail Slots include *Mailboxes*, *Mailing Lists* and *Archives*.

The following topics will be covered in this chapter :—

- The Post Master
- Mailbox Aliases
- Mailboxes
- Mailing Lists and Archives
- Accounting and Mail Groups

An Overview of the Mail Slots

Mailtraq's e-mail server consists of a number of services that work together to move messages around. Mailtraq can only do two things with an e-mail message: deliver it to a local mail slot, or send it to another e-mail server. This section discusses the first of these options.

The mail slots represent the *user* part of the e-mail address (the part before the '@' symbol), and are the names that Mailtraq uses to describe recipient objects. A mail slot can be a *Mailbox* (from which users may collect their mail), a *Mailing List* (for forwarding mail to a number of other recipients), an *Archive* (for storing messages for public access) or a special object. The special objects (of which there is only one of each) are the *Postmaster* mailbox, the Mail Server interface, and the News Server interface. The two interfaces are discussed later, while the Postmaster's mailbox is essentially the same as an ordinary mailbox (except that it cannot be deleted, and has few options).

Ownership

Each mail slot (other than the special objects) has an *owner* and depending on your licensing scheme you may have one or more users which can act as owners. Users can access and configure their mail slots through the *Web Administrator*.

Visibility

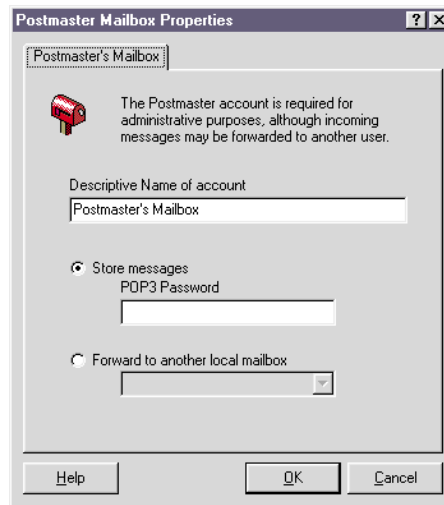
Each mail slot also has a *visibility* option. This determines whether the mail slot appears in the local directory service (which is accessed, again, through the *Web Administrator*). Setting this to *private* means that only the owner can see the mail slot. Setting this to *protected* means that only users who have logged in (and thus local) can see the mail slot, and *public* makes the mail slot visible to everybody.

The Post Master

This special mailbox cannot be deleted, and represents an important function of all mail servers. The Post Master is the name given to the person who is responsible for the administration of e-mail in a domain. It is a requirement that mail to the 'postmaster' at any domain be accepted. Mailtraq will also occasionally generate messages for the postmaster (such as alerts, or reports).

While you can forward messages from the Postmaster to another account, you should ensure that they are read frequently by someone.

*The Postmaster
Mailbox properties
dialogue*



Mailbox Aliases

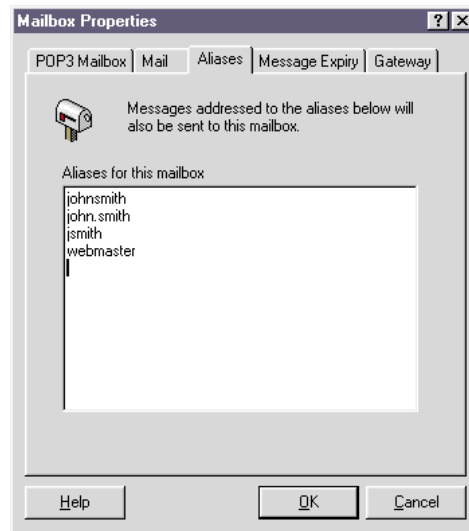
Each mail slot can have a number of aliases. Aliases are alternative names for the slot. For example, they might be alternative spellings to

save confusion and catch errors. Consider the advantage of having the aliases 'philip.thompson, phillip.thompson and phillip.thomson' for the mailbox 'philip.thomson'.

Another example might be taking responsibility for another address, such as having the alias 'webmaster' if it is your responsibility to manage the web pages.

- *Note:* If you have a mailbox called 'webmaster', and have another mailbox with 'webmaster' as an alias, then they will both receive a copy of any mail written to the web master. If not properly managed, this can be a serious security problem. For example, a user simply has to add an alias to watch all mail sent to another mailbox. For this reason, the option to configure the aliases using the Web Administrator can be specified for each user.

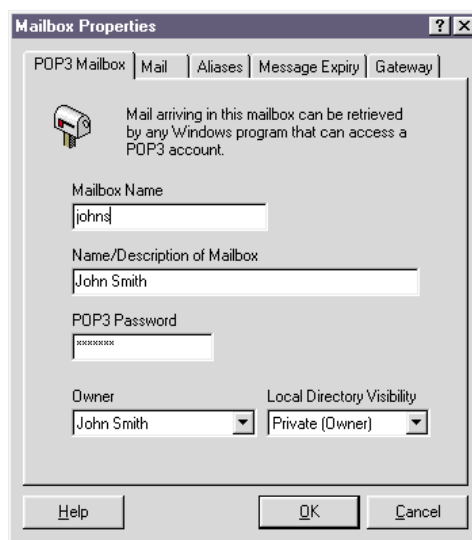
Setting the Aliases for a
Mailbox



Mailboxes

A mailbox is simply a location in to which e-mail messages can be stored. Users may access the mail in their mailboxes from their mail client software (using POP3). Mailboxes also have a number of options that make them very versatile. Mailboxes can forward messages to other addresses, automatically reply to the sender, store a message in an external text file, and automatically delete messages after a specific age.

*The Mailbox Properties
Dialogue*



Accessing Mail from Clients

In order to access the mail in a Mailtraq mailbox, you simply need to tell your mail client the mailbox name, and its password. If you set your mailbox password to be blank, then Mailtraq will accept any password. You can also read mail from within the Mailtraq console by selecting the mailbox (in the object tree) and opening a message. You will see the entire message contents (including the header). You can also read mail using the Web Administrator, which can be useful in situations where you

need to access mail quickly without installing a mail client, or where you are not using your normal machine.

Forwarding Mail

This facility allows a message to be automatically sent to another location. If you select the 'Delete from Mailbox' option, then a copy will not be kept in the mailbox. This may be useful if, for example, a user wants to ensure that a colleague receives a copy of all his e-mail. The mailbox owner may also, for example, wish to have all his mail forwarded to a secretary while he is away.

*Configuring the
Forwarding Options for
a Mailbox*

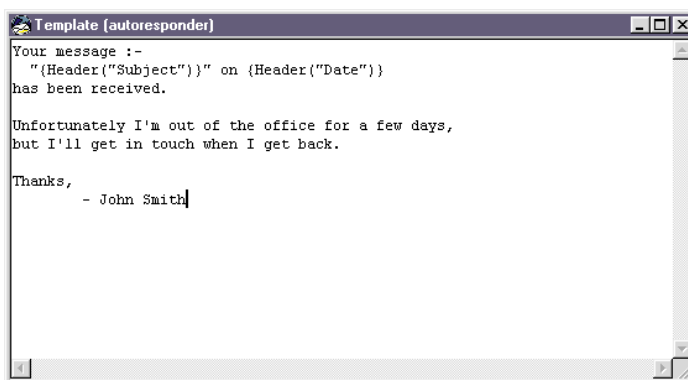


Although it may not be immediately obvious, you may enter more than one e-mail address in the forwarding field. Do keep in mind that, although this may appear to be a simple way to create a mailing list, the Mailing List object does this task far more comprehensively.

Automatic Replies

This facility is also often called an *Autoresponder*. An example might be a vacation message letting the sender know that the recipient will not be able to reply promptly. The response is actually a template (allowing a customised response based upon the actual message being received). Templates are discussed in detail in Chapter 16. Providing you don't use any braces (the '{' and '}' characters) in your automatic response, it is treated as plain text.

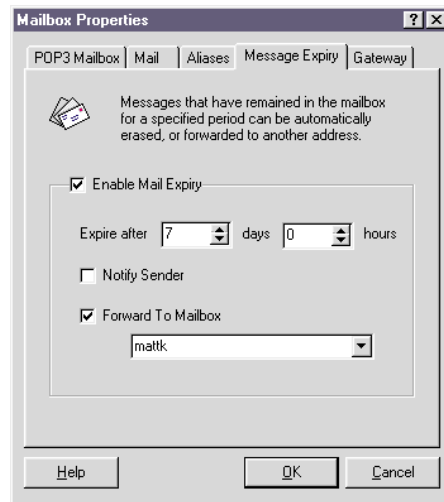
*Entering an
Autoresponder for a
Mailbox*



Message Expiry

This facility simply ensures that old messages are automatically deleted (or sent to another location). If users are given the option to leave mail in their mailbox (even after reading it), you can be sure that many of them will simply never delete their messages. While this may be quite appropriate in some cases, it can also be an abuse of the service.

Configuring Automatic Expiry for a Mailbox

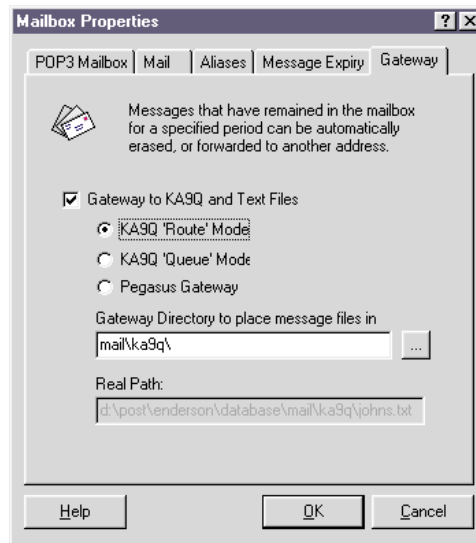


Another use of this facility might be where mail can be re-routed if it hasn't been collected within a specified period.

Mail Gateways

You can either store the message inside the mailbox, or simply output the message to a text file that is external to Mailtraq. This is necessary in order to use mail clients that do not support POP3, or networks that do not implement TCP/IP. This is also an ideal way to output messages for automatic use by other programs.

Configuring a Mailbox Gateway



Mailtraq supports three output formats: *KA9Q Route*, *KA9Q Queue* and *Pegasus*. The *KA9Q Route* option simply appends messages to a file named after the mailbox. The *KA9Q Queue* and *Pegasus* modes create a separate file for each message in the specified directory. The only difference between these two modes is the filename. *KA9Q* files end in *.txt*, while *Pegasus* files end in *.cnm*.

Mailing Lists and Archives

These two types of mail slots share many properties with mailboxes. They accept mail addressed to them, and they can even use aliases to accept mail that is not addressed to them. However, these two objects are discussed in detail in Chapter 11.

*Mailing List Properties
Dialogue*



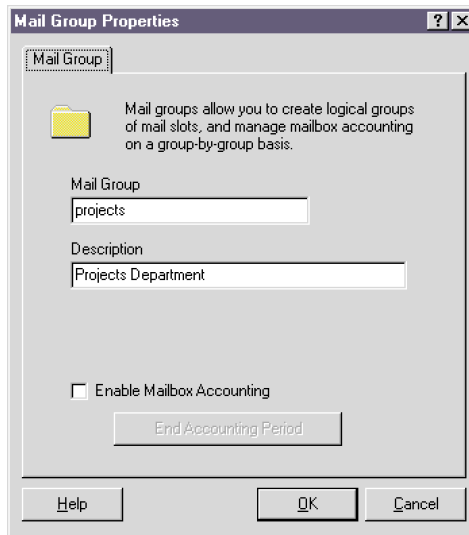
The screenshot shows the 'Mailing List Properties' dialog box with the 'List' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are tabs for 'General', 'Digests', 'Aliases', 'Addresses', 'List', 'Subscribers', 'Access', and 'Messages'. The 'List' tab is active, showing a text area with the following text: 'Mailing lists can be used to distribute messages to subscribers at any address. The maintainer will receive delivery and transport messages.' Below this text are fields for 'List Name' (containing 'projects-dep'), 'Description' (containing 'Projects Department List'), and 'Maintainer' (a dropdown menu showing 'postmaster'). There are two radio buttons: 'Distribute incoming messages to all subscribers' (selected) and 'Distribute evenly between subscribers'. Below these are two more dropdown menus: 'Owner' (showing 'Administrator') and 'Local Directory Visibility' (showing 'Protected (LAN)'). At the bottom are buttons for 'Help', 'OK', and 'Cancel'.

Accounting and Mail Groups

If you manage e-mail for a large organisation, with a large number of users, even the task of finding mail slots can become complex. Once you have more than a few mail slots, it may be easier to group these together in some logical fashion. Mail Groups also have another purpose: they allow you to perform basic accounting tasks.

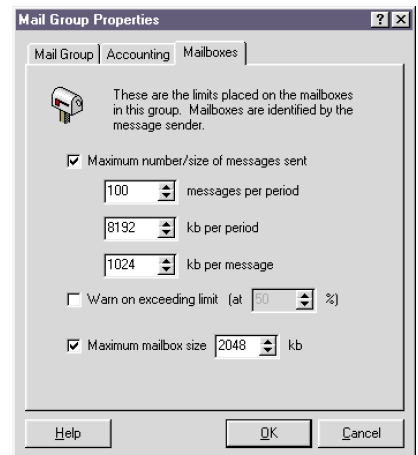
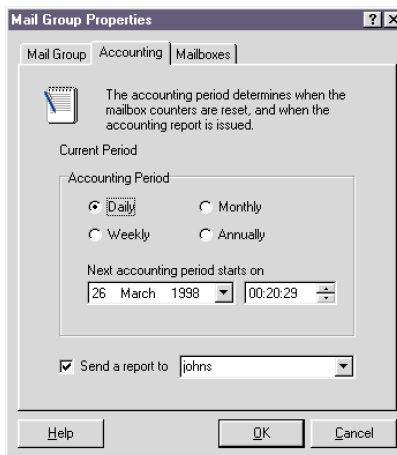
You can create a hierarchy of mail slots, by placing mail groups within other mail groups. Each mail slot is subject to the immediate group in which it is located. Each mail group can impose limits on its mailboxes (such as the number and volume of mail sent out, and the volume of mail stored within the mailbox). It is also possible to keep records on the mailbox usage.

*Mail Group Properties
Dialogue*



If you decide to enable accounting on a mail group, you can have a summary of the mailbox usage for that group sent to an e-mail address at regular periods. This may be an effective way to monitor the use of your system, and to charge people for their use of your system.

*The Mail Group
Accounting Options*



If a message is sent to a mailbox that is approaching its group specified capacity, a message can be sent to the mailbox to warn the owner. When

the capacity is actually reached, no further messages will be stored within it, and messages will be returned to their sender.

8

Mail Delivery

This chapter covers the mail delivery process both within Mailtraq (between Mail Slots) and the delivery of mail between Mailtraq and other Internet mail servers.

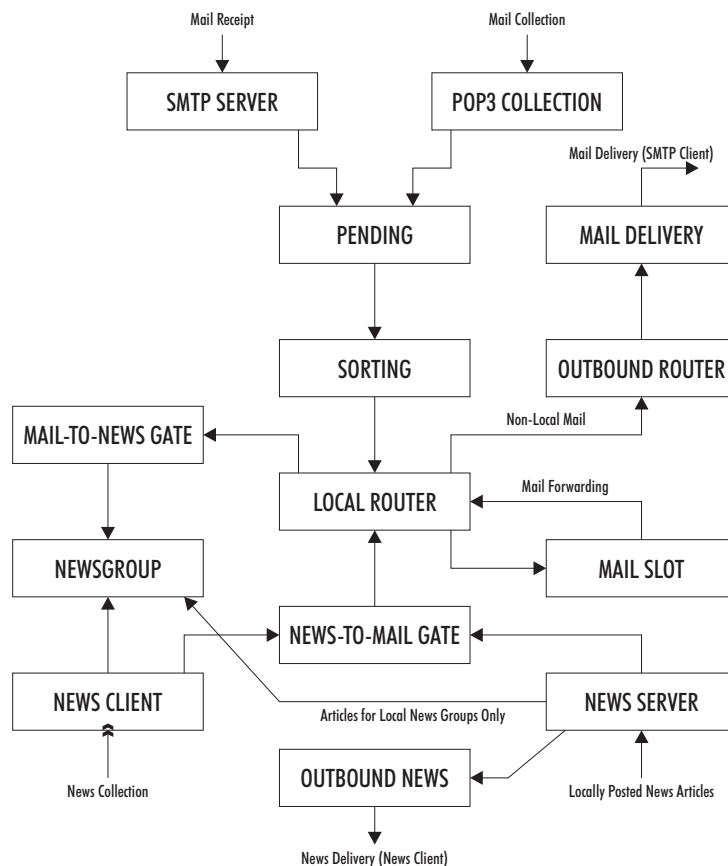
The following topics are covered :—

- An Overview of Mail Delivery
- The Inbound and Outbound Routers
- The Delivery Services (SMTP and POP3)
- Actively Collecting Mail from POP3 servers

Overview of Mail Delivery

As discussed earlier in this manual, there are only two things Mailtraq can do with a message it receives. It can either deliver the message to a mail slot, or to another e-mail server. The part of Mailtraq that makes the decision regarding what happens to a message is called the ‘Router’. There are actually two routers in Mailtraq: the *Inbound Router* and the *Outbound Router*. The illustration shows how these routers fit into the Mailtraq e-mail operation.

The Mail Router showing the flow of messages



But how does e-mail actually get into the router? There are only seven ways for this to take place :—

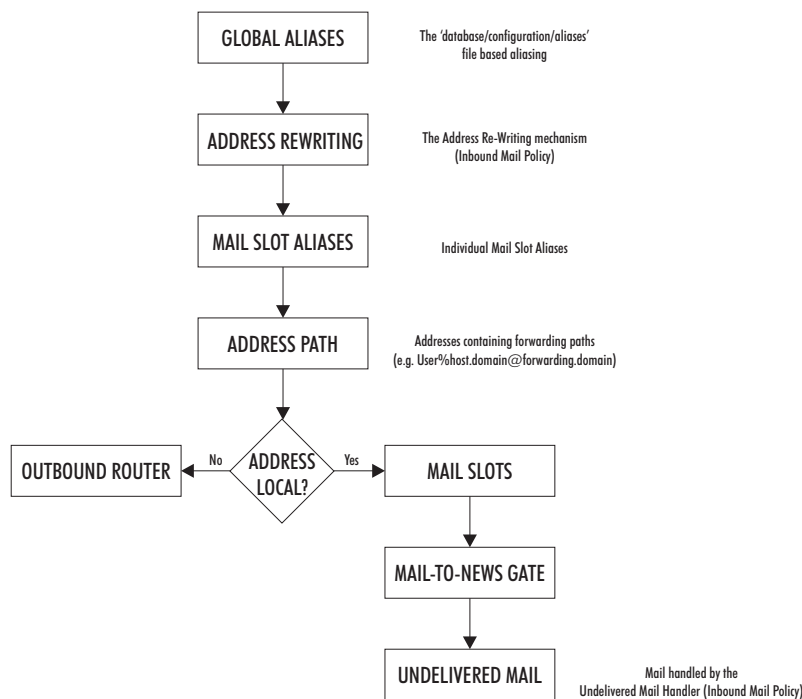
1. Messages are received via SMTP
2. Messages are downloaded via Remote POP3 Collection
3. Messages are imported from a gateway
4. News articles are downloaded and sent through a news-to-mail gateway
5. News articles are posted locally, and sent through the news-to-mail gateway
6. A script generates and posts a new message
7. Messages are inserted directly into the inbound mail queue by some external source

Methods (4) and (5) are discussed in detail in Chapter 10. Methods (6) and (7) are discussed in Chapter 15.

The Inbound Router

Once Mailtraq has received a message, it is placed into the inbound mail queue, and one by one the Inbound Router processes the messages in that queue. This router examines each recipient of a message and decides whether it should be delivered locally, or sent on to the Outbound Router. The Inbound Router considers a number of factors when examining each recipient, as illustrated below.

*The Local Router
showing the flow of
messages*



When messages are sent to other recipients by, for example, mailbox forwarding and mailing lists, these messages re-enter the Inbound Router again. In order to prevent mail loops, no message can enter the Inbound Router more than a specific number of times, after which it is sent to the Post Master. This is configured through

The Inbound Router is listed amongst the Active Connections in the Mailtraq Console. The entry will indicate what message is currently being routed.

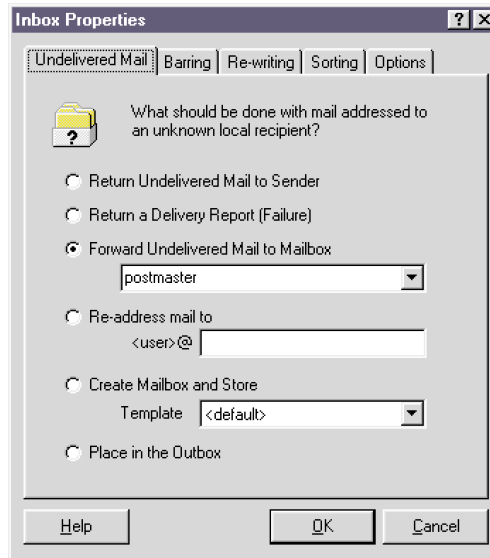
*The Active
Connections window
showing a message
being routed*

Name	Connected To	Status
SMTP Server		<accepting connections>
POP3 Server		<accepting connections>
NNTP Server		<accepting connections>
Mail Router (Out)		Idle
Mail Router (In)		Routing ENDR32452567

Mail Bouncing

This term is used to describe what happens to mail that cannot be delivered to its recipient. The message appears to ‘bounce’ off the mail server, returning to its sender. Mailtraq offers a number of ways to handle bounced mail, and these are configured in the Inbound Mail Policy (illustrated below).

*Handling Undelivered
Mail (Inbound Mail
Policy)*



Return Undelivered Mail to Sender

This option only applies to messages being received with SMTP. Messages received through other methods are actually handled through the “Return a Delivery Report (Failure)”. Part of the SMTP service defines how to deal with mail bounces, and by indicating that the message cannot be accepted before it is received by Mailtraq is certainly the most efficient option available.

The Mail Server that is attempting to send the message is then responsible for providing a delivery report to the sender. What the server actually does is usually very similar to the next option.

Return a Delivery Report (Failure)

When a message that cannot be delivered is processed by the Inbound Router, a report is generated and returned to the message's sender. This delivery report should explain what has happened, although its actual content (since it is a Mailtraq template) is user definable. Delivery Reports are also issued when Mailtraq is attempting to deliver a message via SMTP and receives a mail bounce (as described above).

Forward Undelivered Mail to Mailbox

This option simply sends all mail for which there is no local address to the specified mailbox. While you are first installing Mailtraq, this is possibly a better choice than one of the above (just in case there are any configuration errors).

- *Note* that since the sender is not notified that there was a problem, they will assume that the message was successfully delivered.

Re-Address Mail to...

Use this option if you are operating multiple mail sites, and it is possible that the user to whom the message exists is located at the alternative site. Be careful of doing the same at the alternate site, as if the message returns it will end up in a loop. If you are operating multiple sites, you should consider using either the mail re-writing option, or one of the aliases methods (as the mail can still be bounced normally).

Create a Mailbox and Store

This is a useful way to configure a mail server, as mailboxes can be created simply by receiving normal e-mails. The mailbox will be created by copying the settings from the specified mail slot (so, in reality, it could be any mail slot) before continuing to post the message as normal.

Place in the Outbox

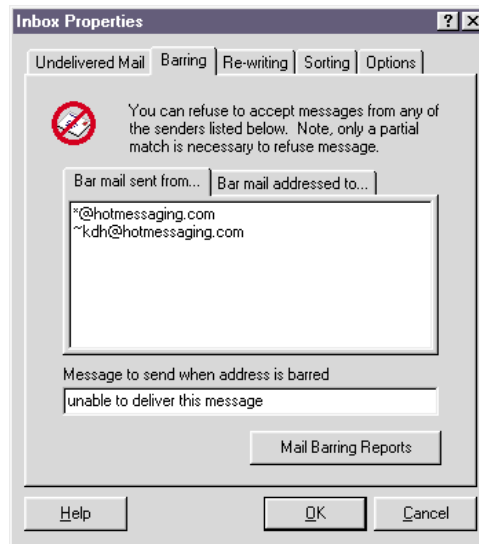
This method should be used in careful conjunction with the static routing tables. The message will then be subject to the normal outbound routing rules. Be careful of creating a mail loop, which is easily done with this option. The outbound routing tables should direct messages to a central

mail server that will not send messages to sub-servers with this option enabled, unless it knows the message won't be bounced.

Mail Barring

There are number of reasons for wanting to refuse to accept certain messages. The most likely reason is an attempt to stop the ever-increasing volume of junk mail on the Internet from wasting your time, and the time of your users. Mailtraq provides the option of refusing messages based on either the sender's e-mail address, or that of the recipients.

*Mail Barring (Inbound
Mail Policy)*



If the message is being received via SMTP, it is possible to prevent the message from being downloaded. A single line reason must be given, and you can configure this to read anything you wish. It may be better to provide as little information as possible, or the sender may take advantage of the response to find another way to send mail. For example, a response such as "We don't accept mail from you" can also be interpreted as "We will only accept mail from you if you don't use that address".

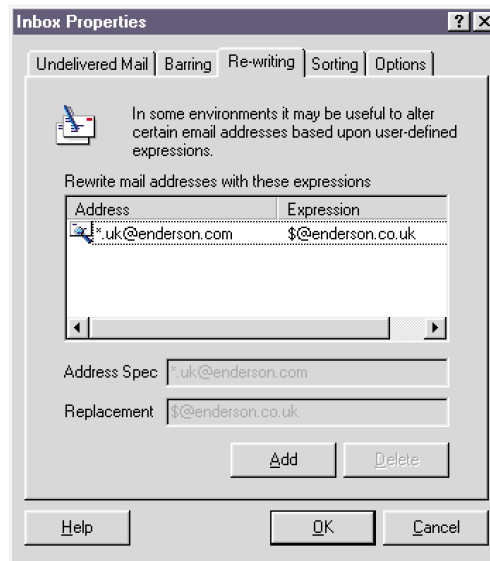
For collection via POP3, it is not possible to return the message to its sender. As Mail Barring is intended as a means to refuse mail, downloading the message and then generating a delivery report defeats the purpose. Thus, barred mail is simply ignored.

- *Note:* for mail receipt via SMTP, the barring applies to the message envelope, not the headers! It is perfectly feasible for the envelope's sender to be different to the *From:* header. When selecting addresses to bar based on mail you have received, you should use the *Return-Path:* header if it is displayed. Alternatively, you can examine the Mailtraq logs to determine the sender's address.

Re-Writing Mail Addresses

Mail Re-writing is an opportunity to change the recipient of a message based upon a set of rules. For example, you might want all mail that was addressed to *enderson1.com* to be sent instead to *enderson2.com*. Another example might be to change all mail addressed to *host.enderson.com* to be sent to *enderson.com*, but with the name *host* appended to the user name.

*Address Re-Writing
(Inbound Mail Policy)*



This is achieved using a series of rules defined using wildcards and placeholders. Each rule consists of a wildcard expression to match the original address, and a replacement expression to provide its new address. The wildcard character '*' represents zero or more characters, while the wildcard character '?' represents a single character.

To use the parts of the original addresses that were matched by the wildcards, you can use the placeholder '\$'. For multiple wildcards, you can use the placeholders \$1 through \$9.

The table below illustrates a few examples :—

Wildcard Expression	Replacement Expression	Example
*@enderson1.com	\$@enderson2.com	john@enderson1.com to john@enderson2.com
*@host.enderson.com	\$.host@enderson.com	john@host.enderson.com to john.host@enderson.com
@.enderson.com	\$1.\$2@enderson.com	john@sales.enderson.com to john.sales@enderson.com

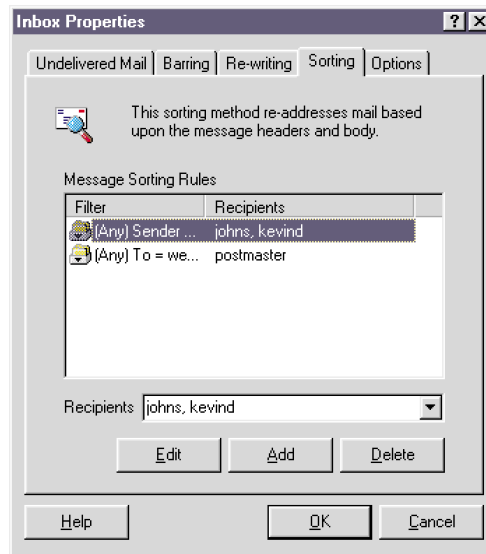
All the rules are applied in order, which means that if an address matches more than one wildcard expression, then both rules will be applied.

- *Note:* since the address re-writing is part of the Inbound Router, it is possible for the same message to have the re-writing rules applied more than once. For example, using the rule `*@enderson.com = $.host@enderson.com` could be applied twice, rewriting `john@enderson.com` to `john.host.host@enderson.com`.

Message Sorting

This facility changes the message recipients based upon message filters. Message filters are able to match messages based upon a set of criteria (such as the message size or subject, or the presence of keywords in the message body).

Message Sorting (Inbound Mail Policy)



Each entry in the sorting table assigns a set of recipient addresses to messages that match the given filter. If a message does not match any of these filters, it is not modified. If the message matches one or more filters, then the recipients will be set to the union of the matching recipients (excluding the original recipients).

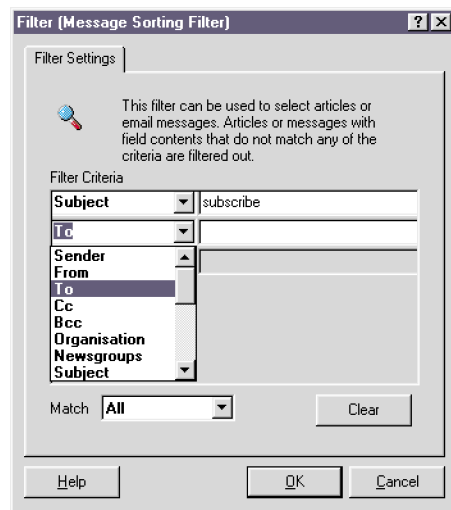
Message Filters

Mailtraq uses 'Message Filters' in several places throughout the system. The filter is a standard mechanism by which certain messages can be included or excluded.

Essentially, the message must pass a set of criteria. Each 'criteria' consists of a field, and a value which is compared with the contents of that field.

Below is an example of a message filter :—

*The Message Filter
Properties*



Each field refers to a ‘Header Field’ from the message being analysed. The drop-down list shows many of the most common header fields, though it does not show all of them, and not all the fields listed will appear in every message.

Fields beginning with the # symbol refer to special properties for the message. These fields are described below :—

#minrcpts	The minimum or maximum number of recipients in the delivery envelope (not the addressees described in the header fields)
#maxrcpts	
#minsize	The minimum or maximum size (in bytes) of the message. This includes the message body.
#maxsize	
#priority	Set to “High” to match high-priority messages
#body	Scan message body for matching text ¹

In the criteria value you can enter any text and Mailtraq will scan the related fields for it. You can also use wildcards (described above).

Neither type of matching is case sensitive.

When multiple criteria are used, there are a number of options to determine how the results of these criteria are combined. There are four options :—

Any	The message will be selected if it passes <i>at least one</i> of the criteria
All	The message will be selected if it passes <i>all</i> the criteria
None	The message will be selected if it does <i>not</i> pass <i>any</i> of the criteria
Not All	The message will be selected if it does <i>not</i> pass <i>all</i> the criteria

When only a single criteria is used, *All* or *None* will determine if the message is *filtered in* or *filtered out*.

Dealing with Large Messages

Some mail servers and mail applications are not equipped to handle large messages efficiently. Large messages also consume large amounts of machine and telephone time, and consequently you may wish to prevent them being downloaded, or break them into smaller portions.

Setting a mail limit will ensure that no message larger than the specified message size enters the Inbound Router. If the 'Break into MIME Partials' option is selected the message will be split into portions using the MIME

- 1 This field can only be used where the message body is available. Filters used during POP3 collection, and News Filtering do not have access to the message body, as only the header is downloaded at the stage when the filter is used.

(Multipurpose Internet Mail Extensions) Partial format. If this option is not selected, then messages of this size will simply not be delivered.

Both the SMTP server and the Remote POP3 Collection facility have mechanisms to prevent the download of large messages, and these are discussed on pages 74 and 114 respectively.

- *Note:* While Mailtraq will split the message, it does not rebuild it. This is the task of the e-mail software used by the message recipient. Unfortunately, not all mail software currently supports MIME Partials. You should ensure that the mail software likely to receive these messages can decode them.

The Outbound Router

The Outbound Router handles mail that is not addressed to the domain that Mailtraq manages. It is possible to see the messages currently in the outbound queue in the Mailtraq Console (in the Outbox object). When a message has been designated for outbound delivery (and its local recipients removed from the message envelope) it is added to the outbound queue.

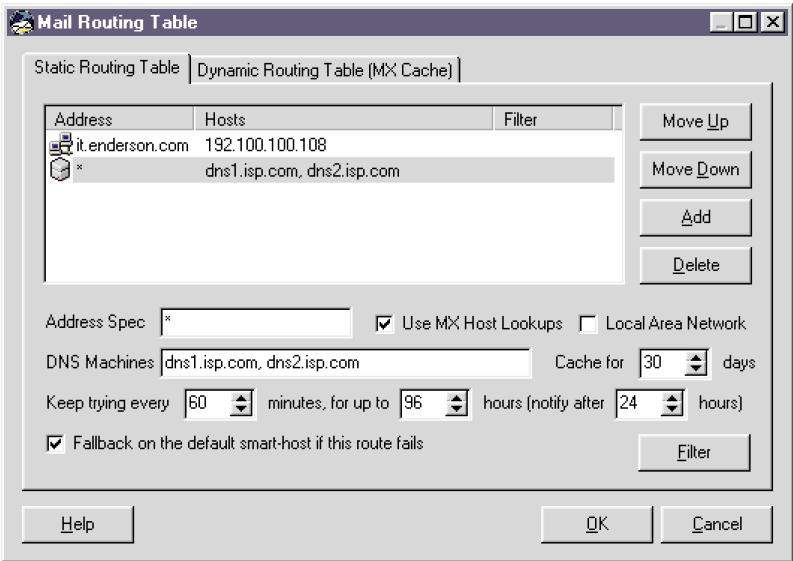
When first added to the queue, the message does not have a route. The Outbound Router regularly scans messages in the queue for those that can be dealt with in some way. If the Outbound Router encounters a message without a route, it scans the static routing table to find the most appropriate route, and assigns it to the message. If the message has

multiple recipients, and the recipients match more than one route, then the message is duplicated once for each route.

Mail Routes

At the heart of the Outbound Router is the routing table (an example is illustrated below). The routing table (also called the static routing table) is a list of user defined routes that messages may take to reach their destination. Each route can either be local or remote, and can either be a list of mail servers, or Domain Name Servers (which can resolve MX requests). Mail can be delivered to machines on local routes without an Internet connection, while messages to remote routes cannot be delivered until Mailtraq is ‘online’.

Static Routing Table
(Advanced Routing
Tables from Outbound
Mail Properties)



Normal routes consist of a list of mail servers to attempt delivery to. MX Routes, however, create the list of mail servers during delivery by obtaining a list of MX hosts from the specified DNS Machines. The MX

Routes are cached in the *Dynamic Routing Table*. It is important to enable the 'Fallback on Smart Host' option for MX Routes, as Mailtraq is not always able to retrieve MX records for a given e-mail address.

Setting the option to fallback on the Smart Host means that the currently configured Smart is implicitly added to the list of mail servers for the route.

Once a mail route has been assigned to a message, that is the only route it can use. It is possible to clear the assigned routes manually through the Mailtraq Console (by selecting 'Clear Host Assignments' from the *Outbox* context menu). During delivery, the messages are tagged with each host in their mail route in turn. Once all routes have been exhausted the message is delayed for the specified period, before the process starts again with the first mail server.

Mail Routes can be configured to issue a delivery delayed warning to the sender once the message has been in the Outbound Router for the specified period. The message can also be canceled, and a delivery report returned to the sender, once the specified terminal period has been reached.

Parallel Mail Delivery

When message delivery commences, a number of separate delivery connections are initiated. At no time will more than the specified number of connections be active. Mailtraq can deliver multiple messages simultaneously to the same server, and will select the optimum set of connections to handle the delivery. Mail delivery is a task that can be greatly improved with parallel delivery. There is always a degree of latency² during the SMTP connection, so delivering more than one

message at the same time makes more efficient use of the bandwidth available.

This can become an even more important issue when sending messages with large numbers of recipients. If a message has a thousand recipients, then if the message is delivered using MX routes it could potentially result in a thousand separately delivered messages. In this case, it would be more efficient to deliver the message to a Smart Host. But how do you configure this?

The Filters (discussed on page 80) can be used to select routes based upon various message criteria. It is possible to select the route only if the number of recipients is below a specified value. The filter criteria would be *#maxrcpt*.

The SMTP Service

The SMTP (Simple Mail Transfer Protocol) is the Internet standard for mail delivery. Most mail clients on your network will probably be sending messages to Mailtraq through the SMTP service, and it is likely that your ISP will be doing so to. You can configure the SMTP service (and Mailtraq's other services) from the *Options* menu in the Mailtraq Console.



Service	Port/Location
Mail Server (SMTP)	25
Mailbox Server (POP3)	110
News Server (NNTP)	119

- 2 The Latency is the amount of time taken for the data to be routed across the network, and for the remote server to process the information.

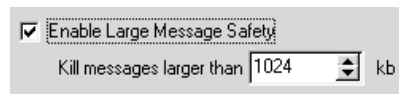
SMTP Services
(Services Dialogue)

Service	Port/Location
Mail Server (SMTP)	25
Mailbox Server (POP3)	110
News Server (NNTP)	119

Although you may have more than one SMTP service (by specifying different ports), other machines will always expect it to be on port 25.

Dealing with Large Messages

One problem with SMTP is that you cannot easily prevent oversized messages from being delivered. It is simply not possible to determine in advance, the size of a message. Therefore, it is not possible to refuse to accept large messages. However, Mailtraq provides a facility that can be used to get around this limitation.



If the connection is cut during download, SMTP clients should try again after a brief period. Mailtraq takes advantage of this by breaking a connection once the message size has exceeded the specified limit, and remembering to refuse the message next time it is offered. This is not a foolproof method, as the only information Mailtraq has to go on is the message envelope. However, since the envelope includes the sender's identity, the worst that can happen is that other messages from that sender will be barred. Mailtraq will only refuse the first message whose envelope matches that of a previous oversized message.

Preventing Duplicate Downloading with POP3

If your Internet Service Provider offers mail via both SMTP and POP3, and you have chosen to fetch it using POP3, then you are likely to receive duplicate messages arriving through SMTP. Since there is little difference between the connections originating from your service provider and other machines, it is not possible to simply have an option in Mailtraq to deal with this. Instead, you must configure a firewall on the SMTP service that denies access to your Internet Service Provider.

As long as you have defined the IP addresses for your local area network (described on page 193), you can simply select the 'Local Area Network Only' firewall option.

The POP3 Service

The Post Office Protocol (version 3) is the most widespread Internet standard for e-mail software to retrieve messages from a mailbox held on a mail server. Mailtraq can provide POP3 access to all its mailboxes, and any authorised mail client that supports POP3 can be used to retrieve mail.

Configuring Mail Clients

Mail Clients will need to be configured with both the name (sometimes called the 'account') and password of the mailbox. The password is set in the mailbox properties (available from the context menus in the Mailtraq Console). If the password is left blank, then Mailtraq will accept any password provided by the mail client.

You should *not* use any of the additional security and authentication methods supported by the mail client. Such facilities may include Secure

Sockets Layer (SSL) and Secure Password Authentication (SPA). You can, however, use Digital Signatures, as these are not affected by the mail delivery mechanisms.

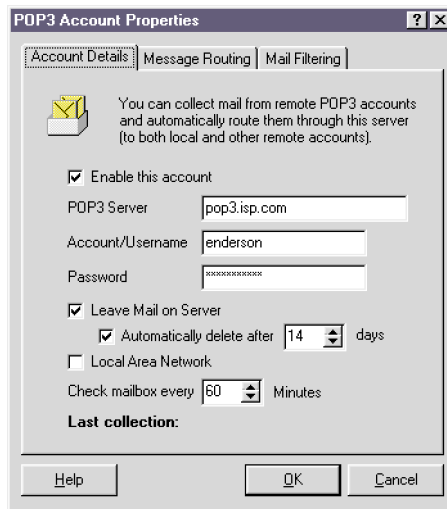
Remote POP3 Collection

This facility allows Mailtraq to retrieve messages from another POP3 server and insert them into the mail router. This can be used as an effective alternative to mail receipt via SMTP, although there are a few disadvantages. During message delivery, the message's envelope is used to determine where it is sent.

Once a message is placed into a POP3 mailbox, the envelope is usually discarded. This can make it very difficult to continue the delivery process when a message is retrieved via POP3. Mailtraq provides a number of ways to determine where the messages are sent to, but it is important to consider these carefully.

The Remote POP3 accounts are configured from the *Options* menu in the Mailtraq Console. Each account represents a connection to a mailbox held on a remote mail server.

POP3 Collection Account Properties



Leaving Mail in the Account

With POP3, deleting a message after it has been downloaded is usually optional. Mailtraq allows you to leave the messages in the mailbox, and only download messages that have not been seen before. Mailtraq can be configured to delete messages after a certain number of days (since they were first seen), which can prevent the mailboxes becoming unmanageable. You should check with your service provider whether or not leaving mail on the server is permitted, and supported.

Leaving mail on the server can be a way to share the account with different users. For example, if you are using an account 'at the office' and at home, leaving messages on the server allows you to access them from both locations.

Routing Messages

Messages that are downloaded can either be sent to a specific address, or they can be routed based upon who the message is addressed to. Since the message envelope is usually lost when the message is stored in the

mailbox, Mailtraq provides a number of options to determine the recipients.

Use these header fields to identify recipients

This option allows Mailtraq to select the message recipients by searching for e-mail addresses in the specified header fields. These fields are listed below (one per line).

Normally Mailtraq would extract the e-mail addresses from the *To:* and *CC:* header fields. However, there is a problem with this method. While these fields indicate to whom the message was written for, they do not necessarily indicate to whom this specific message is addressed. Consider, for example, sending a message to a mailing list. The *To:* field would therefore show the name of the mailing list. Unfortunately, once the message is distributed by that list, the *To:* field continues to point to the list. Clearly sending that message to the contents of the *To:* header is not the required action.

Some service providers add a field to the message header showing the message recipients. If your service provider supports this, then simply enter this field name (and no others) into the list.

Use the *ENV extension

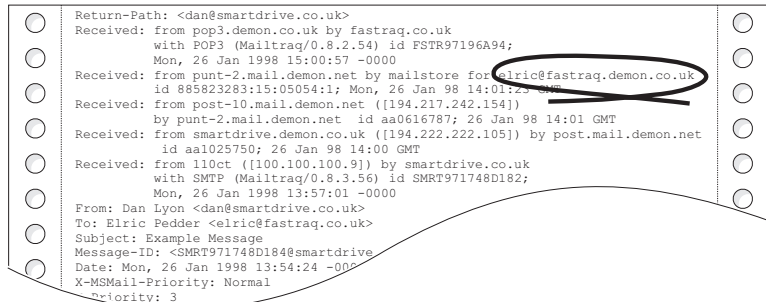
Mailtraq supports the Demon **ENV* extension to POP3, which allows the original envelope to be retrieved. This command is currently only used by a small number of other service providers. If it is available, this is an ideal solution.

- 3 CC stands for Carbon Copy. The original recipients of a message are usually those listed in the *To:* and *CC:* fields.

Extract recipients from *Received:* fields

When a message is delivered between mail servers, each server is required to add a *Received:* header field. Many mail servers include the real recipient of the message in their *Received:* header, and Mailtraq can scan all these fields (starting with the first field added) for this address.

Identifying the recipient from the *Received:* fields



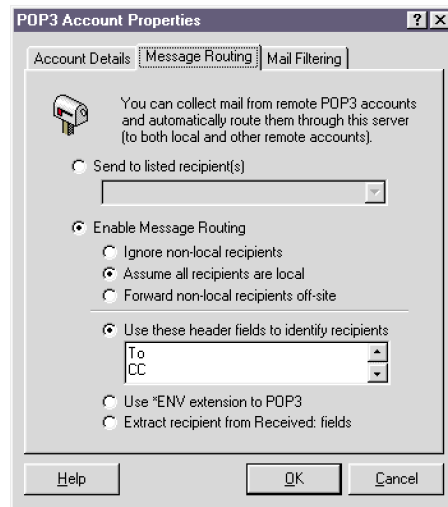
```
Return-Path: <dan@smartdrive.co.uk>
Received: from pop3.demon.co.uk by fastraq.co.uk
with POP3 (Mailtraq/0.8.2.54) id FSTR97196A94;
Mon, 26 Jan 1998 15:00:57 -0000
Received: from punt-2.mail.demon.net by mailstore for elric@fastraq.demon.co.uk
id 885823283:15:05054:1; Mon, 26 Jan 98 14:01:23 -0000
Received: from post-10.mail.demon.net ([194.217.242.154])
by punt-2.mail.demon.net id aa0616787; 26 Jan 98 14:01 GMT
Received: from smartdrive.demon.co.uk ([194.222.222.105]) by post.mail.demon.net
id aal025750; 26 Jan 98 14:00 GMT
Received: from 110oct ([100.100.100.9]) by smartdrive.co.uk
with SMTP (Mailtraq/0.8.3.56) id SMRT971748D182;
Mon, 26 Jan 1998 13:57:01 -0000
From: Dan Lyon <dan@smartdrive.co.uk>
To: Elric Pedder <elric@fastraq.co.uk>
Subject: Example Message
Message-ID: <SMRT971748D184@smartdrive
Date: Mon, 26 Jan 1998 13:54:24 -0000
X-MSMail-Priority: Normal
Priority: 3
```

The above illustration shows how Mailtraq has identified the recipient's address by searching through the *Received:* header fields.

Unfortunately not every mail server will add this information, and where there are multiple recipients, this information cannot be shown.

However, if your service provider's mail servers do add this information, then the presence of this information can be almost guaranteed.

POP3 Collection
Properties for Mail
Routing



Once the e-mail addresses have been extracted, there are further options open.

**Ignore non-local
Recipients**

Firstly, addresses that are not local mail slots can be ignored. This is useful for the addresses extracted from the *To:* header (as it can deal with messages addressed to several people), but it is not appropriate for routed mail (such as messages distributed by mailing lists).

**Assume all recipients
are local**

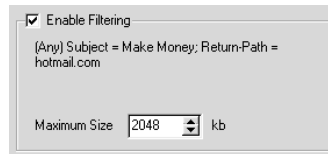
Another option is to assume that recipients are local. This process simply changes the recipient's address to a local address (i.e. *user@some.other.domain* becomes *user@yourdomain.com*). This option does not deal with messages that are addressed to several people, some of which are not local.

**Forward non-local
recipients off-site**

The final option is to forward the non-local recipients off-site. By not modifying the recipient address in any way, the messages are simply routed as normal. If the address does not refer to the local domain, it is placed in the 'outbox' and will be delivered along with other outgoing mail.

Message Filtering

Apart from the opportunity to leave mail on the server, the other important advantage of collection via POP3 is the ability to selectively retrieve messages. You may choose to filter out messages either to leave them for another user to download, or in order to ignore unwanted mail.



If filtering is enabled, then only messages that match the specified filter, and smaller than the specified size, will be automatically retrieved.

Although you may wish to erase messages that are excluded by the filter automatically, it may be more useful to leave them on the server until they are 'expired' by either the server, or by the 'Delete After' setting.

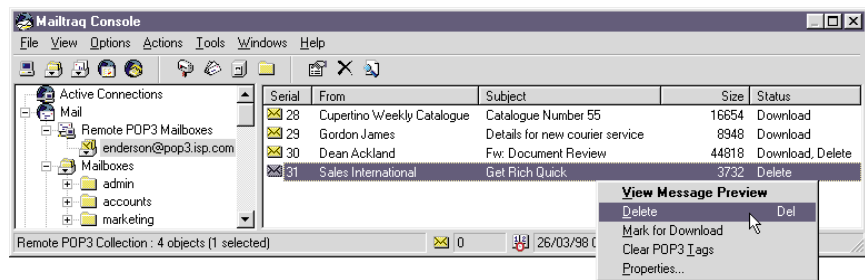
When the POP3 client connects to the server, it first builds an index of the messages available. These are displayed in the Mailtraq Console. Messages that are not automatically downloaded can be tagged for later download. This can be done either by selecting the message, and choosing 'Download' from the context menu, or by using a 'summary' message.

In the POP3 account properties, you may specify that a summary message be posted to an address. The summary consists of a list of all the messages that were not downloaded, with a short preview of each message. The recipient of the summary can reply to the message (the response going to the *mailserver* object) indicating which messages they wish to have downloaded.

The Preview Window

In the Mailtraq Console, when a Remote POP3 account is selected, a list of the messages that were in the mailbox is displayed in the list window. Each message will be tagged according to what was (or will be) done with it. For example, messages that have been downloaded will appear as *fetched*, while messages that will be downloaded during the next collection are tagged as *download*. If you delete a message in this list, it will be marked as *delete*, and during the next visit to that mailbox, the message will be erased.

POP3 Collection Preview Window



9

Administration Topics

This chapter discusses the management and administration of Mailtraq's user accounts and the creation of mail slots, and a number of administrative issues.

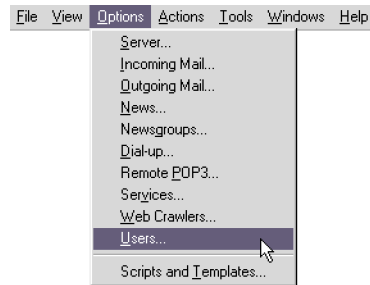
The following topics are covered :—

- Managing Users
- Managing Mail Slots
- Event Logging
- Event Reporting
- The Mailtraq Database

Managing Users

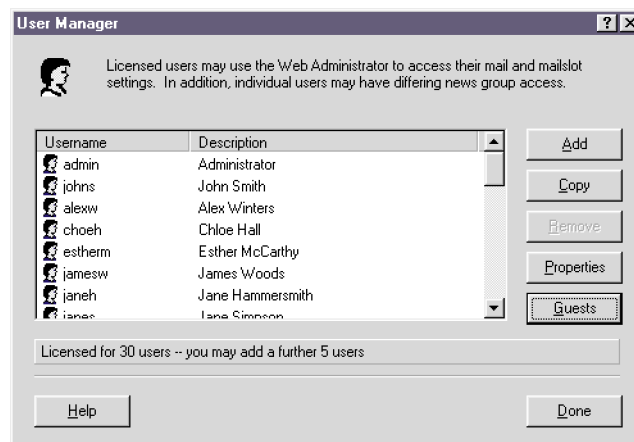
In Mailtraq, a user is someone who uses the services Mailtraq provides, owns mail slots, and can access news groups. The number of users you can have will depend on your licensing scheme.

User accounts are configured from the *Users* entry in the *Options* menu of the Mailtraq Console.



The Users dialogue allows you to add and remove users, based on your licensing scheme, and configure the 'Guest' user account which is used in cases where the current user has not yet (or cannot) log in.

*The User Manager
dialogue*



Users accounts are necessary to provide control over what services and facilities people can access in Mailtraq. Users can control many aspects of their mail slots directly via the Web Administrator.

Specifically, user accounts provide control over :—

- Access to the Web Administrator
- Ownership of mail slots
- Access for reading and posting to news groups
- Use of the Web Proxy facilities

User Account Properties

The properties dialogue allows the user's identifier (username) and their full name to be set, along with a password and the home page. The other tabs allow news group access to be specified, and the privileges when using the Web Administrator to be set.

Configuring the User Properties



The home page is a directory under which web pages and related files can be stored, which can be accessed by using the standard 'homepage' URL:

`http://servername/~username` (where *servername* is the network name of the Mailtraq Host).

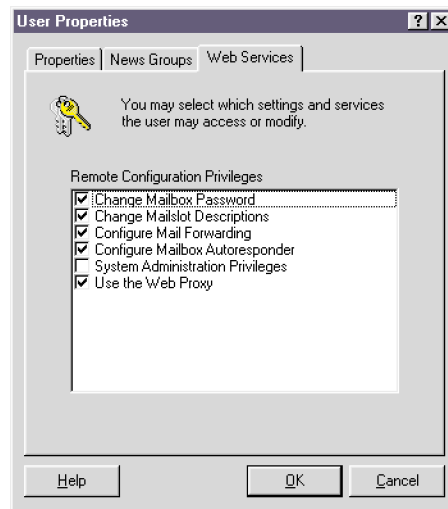
The Web Administrator

See Chapter 13 for details on the Web Administrator

When a user visits the Web Administrator, they will initially be treated as a 'Guest', and consequently be subject to the privileges set in the 'Guest' user account until they log on with their username and password.

When a user visits the 'Directory' option in the Web Administrator, they will be able to see any mail slot with 'Public' visibility. Once a user has logged on, they will also see any 'Protected' mail slot, and they will see any mail slot that they own. They will also be able to modify the properties for their mail slots, based upon the privileges set in the User Properties (shown below).

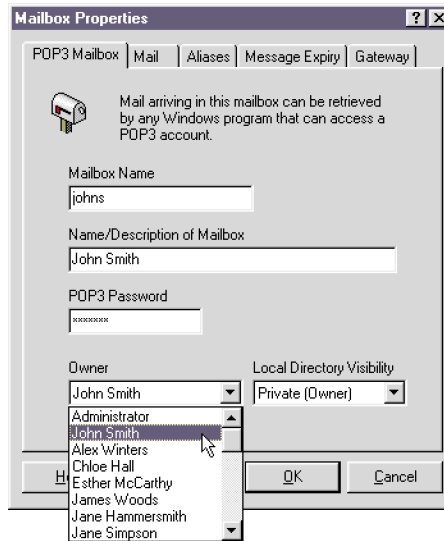
Setting the Web Administrator Privileges



Mail Slot Ownership

Ownership of mail slots can be changed either by dragging them between users in the ‘Users’ part of the Console, or by changing the ‘Owner’ setting in the mail slot properties, as shown below :—

Changing the Mail Slot owner (Mailbox Properties)

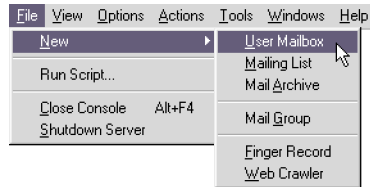


Managing Mail Slots

Each Mail slot appears in the Console under both the ‘Mail’ tree, and the ‘Users’ tree. This allows you to view the mail slot structure from two different perspectives. The mail groups (which provide the hierarchical structure under the ‘Mail’ tree) are used for logical grouping, and to apply the accounting controls to the mail slots they contain.

Creating Mail Slots

To add a new mail slot, or mail group, select the mail group under which you wish to add it, and select the mail slot type from the *New* option in the *File* menu.



You can easily change the mail group in which a mail slot (or group) appears by dragging it between owners.

Importing Mailboxes

If you are migrating from another mail server, or already have a list of mailboxes you wish to create, this can be a quick and effective way to get started.

You will need to have a text file containing a list of at least the mailbox identifiers. Mailtraq can also extract full names and passwords, providing the text file is in a comma-separated format, and each row contains details for one mailbox.

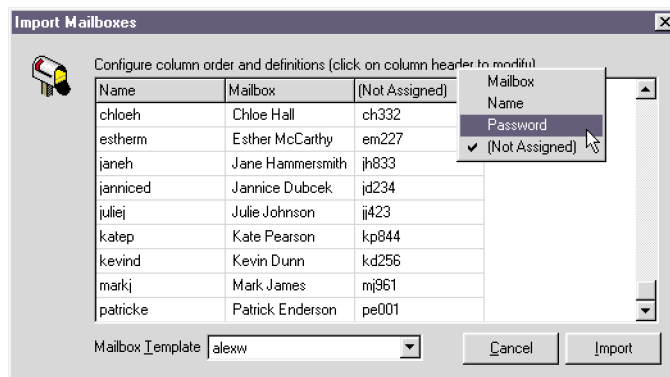
An example is shown below :—

```
johns, John Smith, js321
janes, Jane Simpson, js123
chloeh, Chloe Hall, ch332
estherm, Esther McCarthy, em227
janejh, Jane Hammersmith, jh833
janniced, Jannice Dubcek, jd234
juliej, Julie Johnson, jj423
katep, Kate Pearson, kp844
kevind, Kevin Dunn, kd256
markj, Mark James, mj961
patricke, Patrick Enderson, pe001
```

The first step to create a template mailbox on which the imported mailboxes will be based. You can then begin the import procedure by selecting 'Import Mailboxes' from the *Tools* menu of the Console.

Then select the text file containing the mailbox details. A window similar to the example below will appear, based upon the text file described above, with the columns separated where the commas were found.

*The Import Mailboxes
dialogue*



The label above each column shows how Mailtraq will interpret the values in that column. To change it, simply click on the column and

select the option from the drop down list. Select (*Not Assigned*) to ignore the column contents.

Select the mailbox template on which to base the new mailboxes from the drop down list, and complete the importing process by clicking on the 'Import' button.

Event Logging

As Mailtraq is a very complex system, operating many tasks simultaneously, it is not viable to provide visual cues for everything it does. It is also possible that you will want to examine a past event more closely in order to identify problems or ensure that events took place as expected.

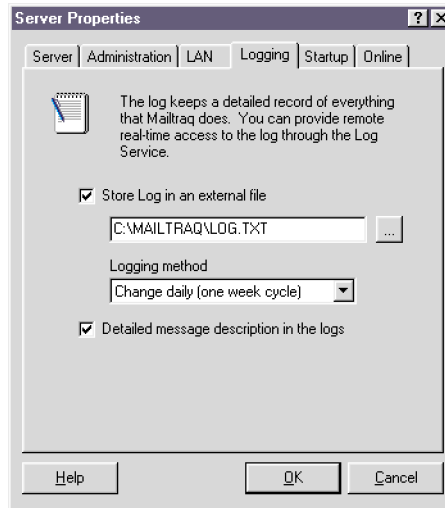
To solve these problems, Mailtraq provides a comprehensive event logging facility. Mailtraq logs many different types of events: everything from transaction messages to mail routing decisions. Every event is listed immediately in the on-screen Event Log, through the remote event logging service, and (optionally) on disk.

Logging events is an essential part of system administration, often providing the only information to explain why something didn't work correctly, and to help correct it. You should enable the on-disk logging even if you don't think it will be an issue, as it will always be after problems occur that you will need the valuable information the logging system provides.

On-Disk Event Logging

This facility is configured in the Server properties (accessed from the *Options* menu in the Mailtraq Console).

Configuring the On-Disk Log (Server Properties Dialogue)



Event Log File

This is the base file name for the disk-based log. If the 'Single File' logging method is chosen, then this will be the only file that will contain the log entries.

Logging Method

The logging method refers to how the log entries are stored in the log files. If you use the single file option, then the file will grow quickly and possibly consume an unnecessary amount of disk space.

An alternative is to rotate the log entries between several files, providing a multi-stage history of past events.

■ Change Daily (one week cycle)

With this option selected, Mailtraq will create seven files — one for each day of the week (based upon the chosen filename). For example, the first event to be logged on Tuesday morning will result in the

Tuesday file being erased and starting only with this event. This option will give you a seven day history

■ Change Daily (one month cycle)

This option is the same as the ‘one week cycle’, except that the log file is changed every day of the month, giving you a maximum 31 day history.

■ Change Monthly (one year cycle)

With this option selected, Mailtraq will create twelve files — one for each month of the year (based upon the chosen filename). This option gives you a one year history, although the log files will be considerably larger.

**Detailed Message
Descriptions in the
logs**

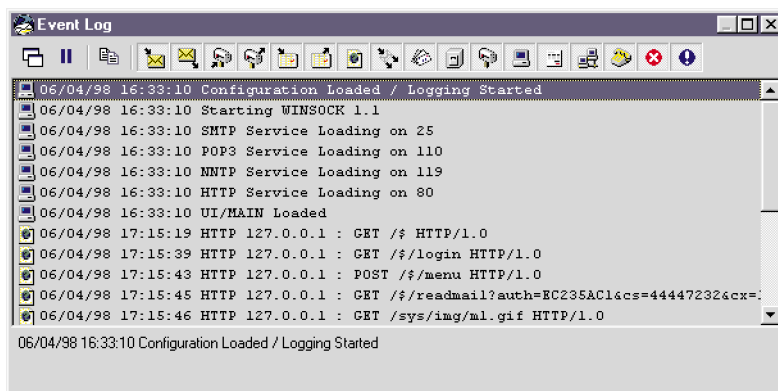
This option affects how messages are referenced in the log entries. Normally, messages are referred to only by their unique Mailtraq ID. It is usually very difficult to cross reference this information, so it is useful only to identify where different log entries refer to the same message.


Enabling this option will result in a detailed reference, including the message sender and subject line. This may be inappropriate for some systems where users do not wish to have the administrators (who may be constantly monitoring the logs) view this information.



**The Event Log
Window**

The Event Log window (available from the Actions menu in the Mailtraq Console) provides essentially the same information as the on-disk log, but on screen and catagorised. It only shows the most recent events, and does not cover any events that took place before the last shutdown.

The Event Log Window



At the top of the Event Log window are a set of toggle buttons that allow you to mask certain categories of event. You can also open multiple log windows with the  button and have a different mask for each window.

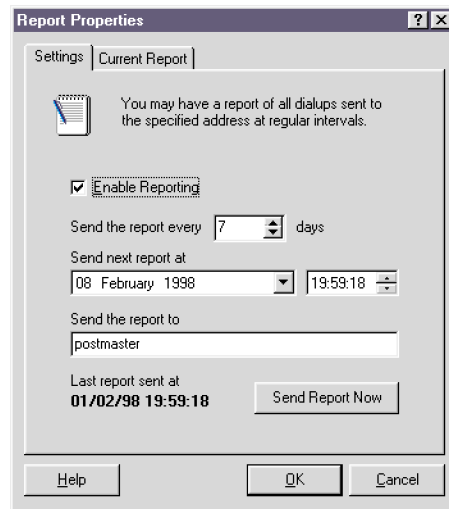
Since the log is continuously being updated, you may have difficulty scrolling through the events while new entries are being added. You can toggle the pause button  to temporarily suspend the addition of new entries. You can also copy the entire log to the Windows clipboard with the  button.

Some entries may be too long to display on the screen. If you click on the entry, the entire text will be displayed at the bottom of the window.

Event Reports

Mailtraq can periodically generate reports on a number of topics, and send them to specific people. Each report is configured in a similar way.

Configuring an Event
Report



The following reports are available :—

■ Dial-up Report

This report includes details on every Internet dial-up, including the amount of time spent on-line and the purpose of the dial-up. The report also provides a sum total of the on-line time for that period.

This is configured in the *Dialup* properties in the *Options* menu.

■ Delivery Failure Report

This report provides information on every delivery failure generated by Mailtraq because it was unable to successfully send a message. The individual failure reports are returned to the message senders, but this report can be sent to an administrator summarising all the failures during the reporting period.

This is configured in the *Outgoing Mail* properties in the *Options* menu.

■ Mail Barring Report

This report contains details on every message that was refused based on the mail barring policy (in the *Incoming Mail* properties — *Options* menu). The report also identifies the barring rule used. The report is also configured in the Incoming Mail Policy.

■ Mailing List Subscription Report

This report contains information of every mailing list subscription and unsubscription, including details on subscribers who were removed by the Address Check messages.

The report is configured from the Mailing List Properties dialogue.

■ Archive Requests Report

This report provides information on all the message requests made to the archive, and is configured in the Archive Properties dialogue.

The Mailtraq Database

The Mailtraq Database contains all the Mailtraq files. This includes the configuration, messages, articles, cache files, and so on. In fact a Mailtraq installation is defined almost entirely by its database. If you move the database to another machine, all the configuration will remain intact. The only information stored outside the database is the user interface

settings and the files (such as the log files) where the user has specified their location.

Database Location

Normally the database will be placed directly below the directory containing the Mailtraq program files. However, if you wish to move the data to another location, you can specify the database path by changing its reference in the Windows Registry.

Mailtraq uses the registry value:

```
HKEY_LOCAL_MACHINE\Software\Fastraq\Mailtraq
DatabasePath=C:\Program Files\Mailtraq
```

to locate the database during startup. The path stored in this registry value is the directory under which the 'database' directory can be found. For example, if the registry value contained "c:\program files\mailtraq", then the database would be stored in "c:\program files\mailtraq\database".

You can use the *REGEDIT.EXE* program (which is part of Windows) to edit the registry values.

The files in the database should never be edited whilst Mailtraq is running. You can, however, insert messages directly into the inbound message router by placing them in the *database\mail\pending* directory. You must use the Mailtraq message format (described below).

Network Drives

You should avoid placing Mailtraq's database on a network drive. Firstly, network drives are almost always considerably slower, and Mailtraq will generate a large amount of network traffic while it is operating. Secondly, if Mailtraq is started as a service it is likely to start before the network

drives are available. If Mailtraq is started before the network drives are ready, it will not be able to load successfully.

Mailtraq Message Format

Throughout the database, messages will be stored by Mailtraq. Essentially, the format is the combination of the message envelope and the message itself. They always follow the same format:

The first line must begin with “FROM: ”, followed by the e-mail address of the sender.

The second line must begin with “RCPT: ”, followed by the e-mail address of a recipient. This line must be repeated once for every recipient of the message.

After the message envelope, the message itself (header fields then body separated by a blank line) follows. Here is an example :—

```
FROM: johns@enderson.com
RCPT: janes@enderson.com
RCPT: alex@widgetmakers.com
Subject: New widget
From: John Smith <johns@enderson.com>
To: Alex Johnson <alex@widgetmakers.com>
CC: Jane Simpson <janes@enderson.com>
Date: Tue, April 07 1998 15:15:50 GMT
```

Have you got the new widget specification yet?

--

John Smith

- *Note:* do not edit the database files ending in ‘.AFV’. Although they appear use the standard message format, other files reference the precise position that each message starts at.

- *Note:* never access the files whilst Mailtraq is running — while accessing them you may prevent Mailtraq from writing to them, resulting in lost data.

10

The News System

This chapter introduces the Internet News system, and discusses how Mailtraq uses it. This chapter also deals with Mailtraq's own news services and how they are integrated with the e-mail system.

The following topics are covered :—

- Introduction to Internet News
- News Collection
- News and Mail Gateways
- The News Server

Introduction to Internet News

The Internet News system is almost as old as e-mail and there is actually very little difference between the two types of communication. A news article looks very similar to an e-mail message: they both have the same header and body structure, and they are in fact both defined by the same Internet Standard. The difference is in the way the messages are delivered.

The Internet News network is also known by its original name: The USENET. This is a web of agreements between service providers to deliver news articles to each other. When the USENET started, this was a small and manageable quantity of information, providing an effective means of group communication. Now the USENET carries well over 45 Gigabytes of articles daily.

The USENET is made up of news groups (over 25,000), covering various specific subjects. Each News Service Provider will host many of these groups, and will decide how long to store articles for each group. Since it is usually unfeasible for users to download the entire USENET, the News Servers supply individual articles on demand to News Clients. Normally, a News Client downloads an overview of all the articles in a group, and then the user selects the articles they wish to read. This works particularly well if the user is on-line at the time, but this can be an expensive approach. Some software allows the user to mark which articles they wish to read, and then downloads them in a single batch when the user next goes online.

While these approaches work well for single user environments, they are less effective for networks where many users read the same groups. Having to download articles in batches can also hinder reading, especially if the user is likely to read a large number of articles from a group.

Mailtraq's approach is to download all the available articles from subscribed news groups as efficiently as possible. Since Mailtraq does not first download an overview of the articles, and because it can download articles from several groups in parallel, it is more efficient than most news clients. Another advantage is that Mailtraq can download articles whenever it is online, taking advantage of connections that are initiated for mail or web activity.

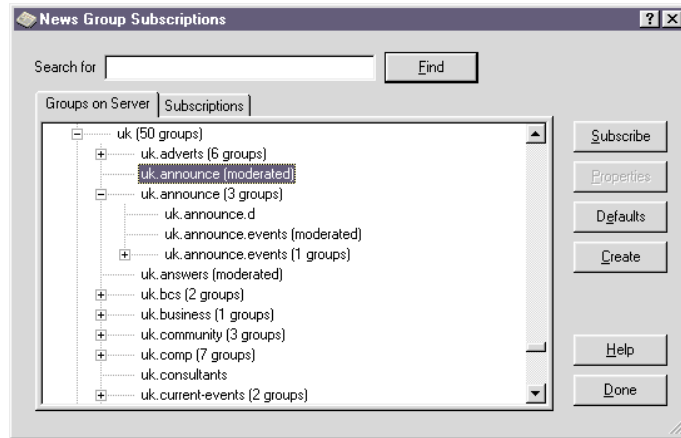
News clients can then connect to Mailtraq instead of the normal news providers, as Mailtraq provides a fully functional News Server. In a multi-user environment, Mailtraq adds further advantages to this approach to handling news. For example, the administrator can control exactly which news groups specific users can read and post to. Further, local news groups can be created which are not official USENET groups, and as such do not appear outside the Mailtraq network. This can be a useful tool for company-wide discussion forums and notice boards.

News Collection

The first step in configuring a news system is to decide which groups to download. Mailtraq can download articles from any number of news servers, but each group can be downloaded only from one server. There are two ways to subscribe to a group: enter the name explicitly, or search

the list of groups available on the news servers. In order to search the list, a group list must first be downloaded from the server.

Subscribing to a News Group

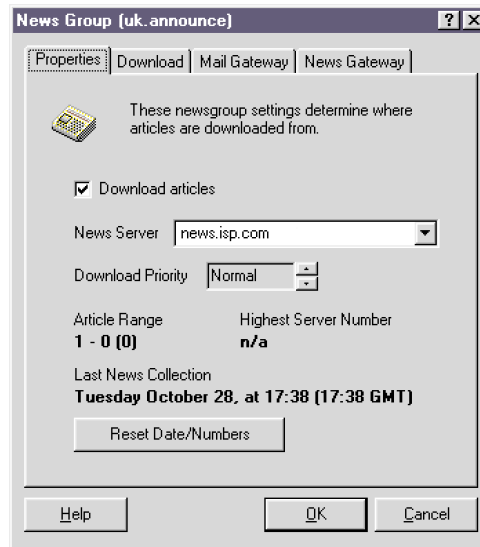


You can see the current subscriptions and available groups by selecting Subscriptions from the *Options* menu in the Mailtraq Console. If the list of groups is empty (even if you have configured a news server) then the list must first be downloaded (which, if the option is checked in the news server properties, will take place on the next connection).

News Groups

Once you have subscribed to some news groups, you can further configure how news will be downloaded. This is done in the Group Properties dialogue.

*News Group Properties
dialogue*



*Connections can be
safely cut midway
through download*

You do not need to wait for the news group downloads to complete before going offline, and if a connection is cut while an article is being downloaded, no harm will be done. This makes it useful to download groups that are more important before others. Each group can have its 'priority' configured, and this will determine the order in which groups are downloaded (high priority groups are downloaded first).

You can also configure which articles are downloaded, in the 'Download' tab of the News Group Properties. When you first download articles from a group, Mailtraq may have to do some catching up. If it is not necessary to download very old articles, a sensible limit should be set on the age of the articles. Specifying how long to store the articles determines when articles are erased (expired) from the Mailtraq database. If this value is set too high, then the Mailtraq database may become unnecessarily large.

The article filter can be a useful way to select, or exclude, articles. If a maximum line length is selected, or if the filter is enabled, then Mailtraq

will first request the article header, then its body (rather than making a single request). This is slightly less efficient, but is worthwhile if you plan to exclude a reasonable number of articles. Note, however, that the filter can apply only to the message headers (i.e. the *#body* filter criteria will never match anything).

News Servers

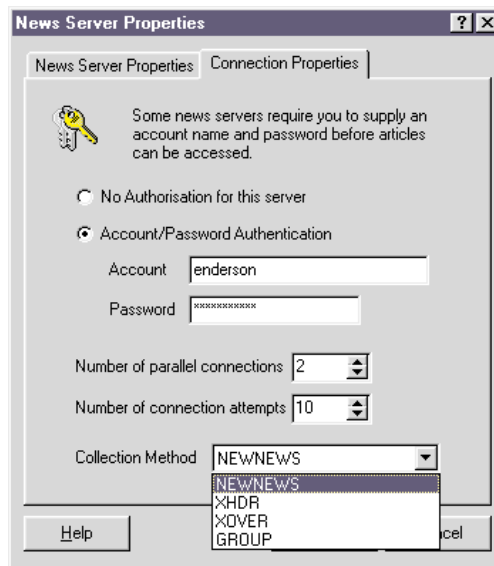
You may also configure how articles are downloaded from servers. Mailtraq allows you to specify the maximum number of parallel connections to use when downloading from a specific group. This facility considerably improves the performance of news download. A fair portion of the time spent downloading articles is actually taken by the news server searching for the requested article. This adds to the 'latency' of the connection. By having Mailtraq download several articles at once, the bandwidth available to your Internet connection is used more efficiently.

- *Note* that the more connections are used, the more memory is consumed by Mailtraq. Additionally, less bandwidth becomes available for other connections (such as mail and web), so a sensible balance should be selected.

The Collection Method is another important setting. Before Mailtraq downloads articles from a news group, it must determine which articles to download. It does this by obtaining a list of the articles, and making its selection from that. Since Mailtraq is essentially trying to keep up-to-date, the best method is to simply ask for a list of articles that have arrived since it last checked. This is called the *NEWNEWS* method. However, this is not the most efficient method for many news clients (who need enough information to allow the user to select articles). This is usually done with the *XOVER* command, which sends an overview of the

articles. Consequently, many News Servers are optimised for the *XOVER* command. If a server is not optimised for *NEWNEWS*, it can be a great resource drain. For this reason, some servers do not permit clients to use that command.

*News Server Properties
dialogue*



Mailtraq allows you to select the collection method, since different news servers will provide better performance for some methods, but not others. You can use the table below to select the method to use.

	NEWNEWS	XHDR	XOVER	GROUP
Reliability	Excellent	Average	Average	Poor
Article Duplication	Excellent	Excellent	Excellent	Poor
Server Efficiency	Varies	Varies	Excellent	Excellent
Bandwidth Used	Good	Good	Poor	Excellent

You can use the event logs to see if the command is supported on your news server

We recommend using the `NEWNEWS` method unless it is not supported. The `NEWNEWS` and `XHDR` commands return a list of articles. The `XOVER` command returns additional information for each article, all of which is discarded. The `GROUP` command is somewhat misleading, as in this case no command is used at all. Mailtraq simply requests articles based upon their number (knowing what the last article number it downloaded was).

The Default News Server

When an article is posted by a user with Mailtraq, it determines which server it should be uploaded to by cross-checking the recipient groups with the current subscriptions. If necessary, multiple copies of the message will be uploaded. If the article is posted to a group that is not currently subscribed to, then Mailtraq will send the article to the default news server.

News and Mail Gateways

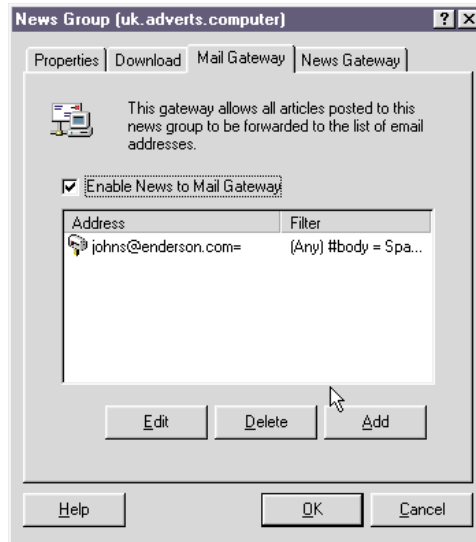
Because news articles and e-mail messages are very similar, Mailtraq takes advantage of this by providing the means to integrate the two types of messaging. Gateways allow e-mail messages to become news articles, and vice versa.

News-To-Mail Gateways

This facility allows Mailtraq to post articles to users in the form of e-mail. Each gateway can filter out messages they are not interested in, and this can be a useful way to ‘watch’ a news group. For example, you may wish to receive (via e-mail) any message posted to `uk.adverts` with the word

‘SparcStation’ appearing in it. This is certainly easier than reading all the articles each day.

Configuring
News-To-Mail
Gateways (News
Group Properties)



Another important use for *News-To-Mail* gateways is to mirror a mailing list, and this is described shortly.

Each gateway posts messages to the specified e-mail address, but only if they pass the filter. Unlike the filter on article downloading, the entire message is available (allowing the use of the *#body* and *#size* criteria, for example).

*Configuring a
News-To-Mail
Gateway*



Only messages that are downloaded (or posted locally) are posted to these gateways. Messages posted from Mail-To-News gateways are excluded (otherwise, it would not be possible to mirror mailing lists).

**Mail-To-News
Gateways**

This gateway allows users to post messages to news groups as though they were e-mail addresses. The e-mail address is made up from the news group name and the Mailtraq domain. For example, at the familiar Enderson Enterprises the newsgroup *uk.adverts* would be *uk.adverts@enderson.com*.

The News-To-Mail
Gateway tab (News
Group Properties)



The *newsserver* Object

This special mail slot is another gateway to convert e-mail messages to news articles. Unlike the normal Mail-To-News gateways, the *newsserver* will forward mail for any news group. You will need to use the *Newsgroups:* header, in the same way as an ordinary news article. Multiple groups can be specified by separating them with commas.

Both the Mail-To-News and *newsserver* gateways cannot authenticate the article senders. Therefore, only the news groups specified in the default user settings can be posted to.

Mailing List Mirroring

There is a particularly important aspect common to both news groups and discussion forums: users frequently reply to each other's messages, creating 'discussion threads'. Almost every news client takes advantage of this by grouping threads together in some way, greatly enhancing the readability of the news group. However, this is often a feature left out of mail clients, and many news clients cannot process e-mail.

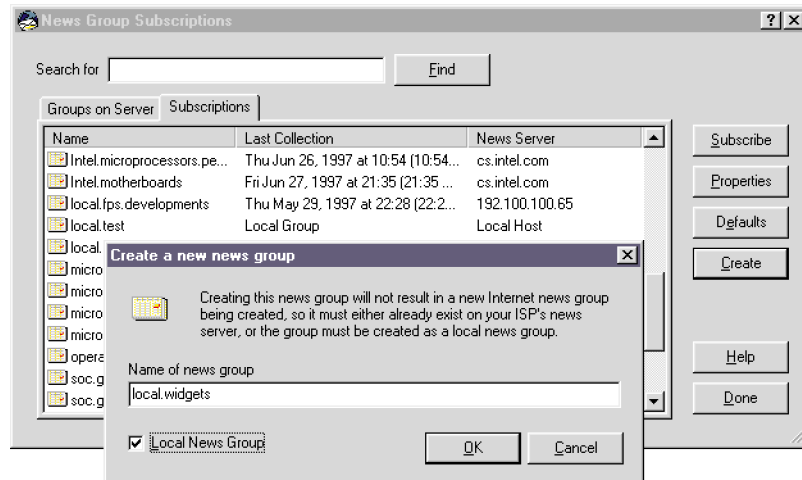
Mailtraq provides an effective solution by allowing a mailing list to be mirrored in a local news group. This has other advantages too. For example, all users on a local network can access the mailing list whenever they wish, reducing the volume of mail to be processed, and allowing users to read articles available before they subscribed.

Mailtraq can achieve this with a combination of News-To-Mail and Mail-To-News gateways. Mailtraq will ensure that messages do not go through both gateways (and thus create a loop).

The example below shows how the mailing list *widgets@widgetmakers.com* can be mirrored as a local news group (local.widgets) :—

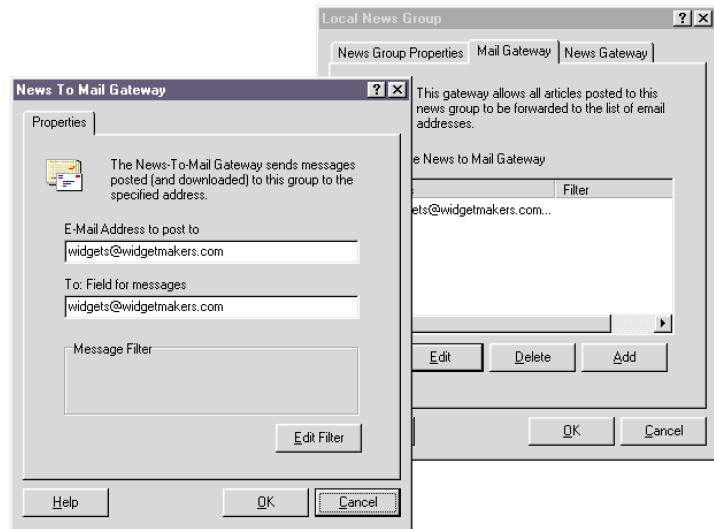
1. Create the local news group, calling it 'local.widgets'

Creating a New Local News Group



2. Add a *News-To-Mail* gateway that sends messages to 'widgets@widgetmakers.com' (also setting to *To:* field to the same address)

Adding a
News-To-Mail
Gateway



3. Enable the *Mail-To-News* gateway
4. Subscribe to the mailing list under the name of the *Mail-To-News* gateway (i.e. 'local.widgets@enderson.com')

You will also need to ensure that everyone who will be posting to the news group is also permitted to post to the mailing list.

You should set the *To:* field in the *News-To-Mail* gateway to the same address as the mailing list in order to let Mailtraq rewrite that header field. Some mailing lists will not accept messages that aren't addressed to them (to avoid loops and mis-routed mail).

Now you can subscribe to the news group ‘local.widgets’ just as any other group, but with the advantage that messages will be better managed by the news client.

The Mailtraq News Server

The news server allows local news clients (news readers) to connect to the news groups downloaded and the local news groups managed by Mailtraq. The news clients can also post news articles to the news server, and Mailtraq will either store them in local groups or forward them to other news servers.

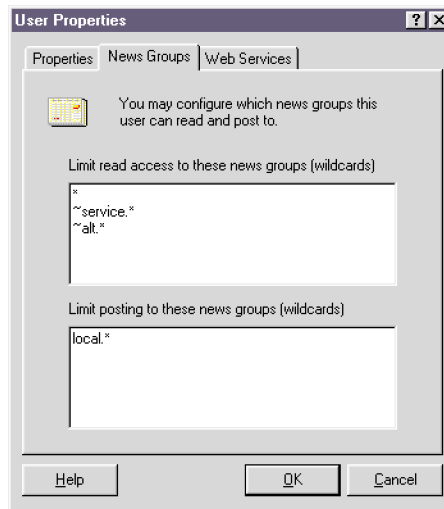
The NNTP (Network News Transfer Protocol) service provides access to the news server for the news clients, and can be configured in the Services dialogue (accessed from the *Options* menu in the Mailtraq Console).

Authentication

News readers will normally have access to the news groups listed in the properties for the ‘Guest’ user (*Users* in the *Options* menu). If the news reader supports user/password authentication (very few news readers do not) then the user can log on to their ‘user’ account and access the news groups listed there.

The illustration below shows the User Properties for the Guests :—

*Setting News Group
access in the User
Properties dialogue*



The standard wildcard list format is used (see page 77 for more information on wildcards). By prefixing a line with a tilde (~) it can be excluded from the list. In the above example, the users are able to read any groups, except those starting with 'service.' and those starting with 'alt.'

Posting News

Users may post articles with their news readers (though only to permitted groups) and the articles will be placed in the *Outgoing News* folder of the Mailtraq Console. If the news groups to which the articles are sent have been subscribed to, then the articles will be sent to the news server from which they are downloaded — otherwise they will be sent to the default news server.

11

Mailing Lists

Mailing Lists are one of Mailtraq's most powerful facilities, providing functionality that would usually only be found in expensive specialised mailing list software. This chapter discusses the use of mailing lists, their administration, and the related public archive facilities.

The following topics are covered :—

- What is a Mailing List
- Mailing Lists in Mailtraq
- Subscription by Mail
- Message Templates
- Public Archives

What is a Mailing List?

Mailing lists are a very effective means of group communication, with a variety of applications. Essentially, a mailing list simply receives a message, and then forwards it on to a number of recipients (subscribers). Mailing lists are described in the same way as any user, with an e-mail address. For example, you might have the list *sales@enderson.com* to handle sales inquiries from the public, or *widget@enderson.com* to discuss the project development.

Distribution Lists allow users to refer to a group of people through a single address

The above two examples actually illustrate two distinctive applications of mailing lists. In the *sales@enderson.com* list, anybody in the world can post a message to the list. The author of the message has no idea who will actually receive it (or even if it is a mailing list at all), and they don't need to know. This level of abstraction can be very useful, as if the person (or people) responsible for sales change, you won't need to issue a new address to everyone. You may choose to have the Sales mailing list sent to several people, or perhaps to one person at a time (spreading the volume of mail between everyone on the sales team).

Discussion Forums allow subscribers to both receive and submit messages to the list

These are all examples of a distribution list. Another type of mailing list is the 'discussion forum'. An example of this is the *widget@enderson.com* list. Here, the subscribers write messages to the list, and all subscribers receive the messages. This is very similar to a news group, but with the advantage of being able to control membership and management of the list. A closed forum allows only the subscribers to post messages to the mailing list, allowing the administrators to decide whom to admit.

What is Subscription by mail?

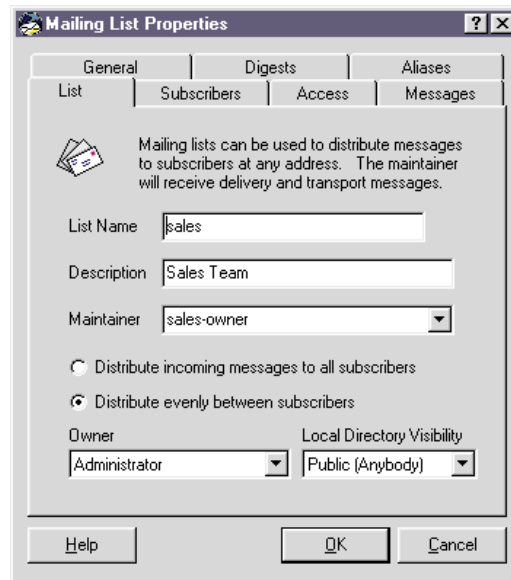
Many list systems (including Mailtraq) allow users to become subscribers through an automatic mechanism, such as subscription-by-mail. With a facility such as this in place, very large mailing lists can become quite easy to maintain.

Using subscription-by-mail, a user can write to the mailing list server with a special ‘control’ message, and using the information provided in that message (such as the user’s name and e-mail address), they can be added to the list. Users can also leave the mailing list by sending a similar message.

Mailing Lists in Mailtraq

Mailtraq provides most of the mailing list facilities found in dedicated list managers. Each mailing list is represented by a mail slot (in a similar way to a normal mailbox) and is subject to the same message routing mechanisms.

*Mailing List Properties
Dialogue*



When a message is received by a mailing list, the message itself, and the mailing list configuration, determine what is done with it. If the message is not admitted, a notice may be returned to the user, or (in the case of moderated lists) it may be sent to the moderator. If the message is accepted, it will be sent to all the appropriate subscribers. Mailtraq does this by creating a duplicate of the original message, and changing the envelope recipients. Mailtraq may also (depending on the configuration) alter various header fields as appropriate.

The result is a single message, addressed to a number of people, which is why you will see only one message in the 'outbox'. During the delivery of the message, Mailtraq may split it into multiple messages, depending on how outbound mail is routed. As discussed on page 109, you may enable MX resolution (where messages are sent to the recipient's designated mail host). Without MX resolution, a single message will be sent to your smart-host, but with MX resolution enabled, Mailtraq may have to deliver

a large number of messages. Usually this is comparatively inefficient (depending on the mail hosts involved), and can result in more time being spent on the delivery process.

The List Maintainer

What happens when a subscriber closes their e-mail account without first leaving the mailing list? The correct procedure is for the mail server responsible for their account to return a delivery failure report. This is usually sent to the message envelope's 'sender'. If the message were sent from the mailing list, you would expect the mailing list to be the sender. However, this would mean that the delivery reports would be sent back to the list, and distributed to all subscribers again (creating a mail loop which is undetectable by Mailtraq).

In order to avoid this, Mailtraq lists have a 'Maintainer'. This is the address shown in the message envelope's sender field, and this is the address to which all errors and reports are sent. It is common practice to use the name of the list with the suffix '-owner' as the maintainer, which makes the message easily identifiable from only its envelope.

Even Distribution

Consider the sales mailing list mentioned earlier. Perhaps Enderson Enterprises has a large number of sales inquiries, and wishes to ease the workload on the person who answers the e-mail. Instead of sending all the messages to the same people, they can spread the incoming messages to one subscriber at a time. For example, if the list had three subscribers and a dozen messages were sent to the list, then each subscriber would receive only four.

Mailtraq does this on a 'round-robin' scheme, but it is not necessarily predictable. In Mailtraq, each subscriber may have a filter, and Mailtraq

selects each successive recipient from the list of subscribers after they have been filtered. If a number of filters are in use, the distribution may appear random (with subscribers possibly receiving more than one message in succession).

The Subscribers

Each subscriber has a number of 'properties'. Their name is used as a display in the subscriber lists, but is not used anywhere else. Subscribers also have two e-mail addresses. The normal subscription (or *Sender*) address is the address used to identify them. In other words, this is the address that they must post from in order for the mailing list to recognise them (and decide whether to accept their message).

The other e-mail address (or *Reply-To* address) is the address to which messages from the mailing list are sent. Note that this address is used to send messages that are being distributed, not messages specific to the subscriber. For example, if the subscriber requests a status report (as part of the subscription by mail facility), then it is sent to their normal address, not the *Reply-To* address.

Subscribers also have a 'details' field, which can contain any short piece of information about them. When Mailtraq creates the subscription itself, it will use the contents of the 'Organization' header field (if it is available). If the list is 'secure' then each subscriber can also have a password.

The subscriber flags are a set of on/off settings for a number of features. The on-line help will provide further specific details on these flags.

The guest subscriber (accessed through 'Guest Properties' on the Access Tab) represents any user posting to the mailing list that is not currently a

subscriber. The ‘New Subscriber Properties’ are used to define the settings assigned to newly created subscriptions.

Moderating Messages

There are two ways to control the messages being posted to the mailing list:

- Limit posting to specific people only,
- and Moderation

Moderation simply means that a moderator has to check every message before it is distributed by the list. In Mailtraq, each subscriber (and guests) can have their messages moderated, or automatically accepted.

Each mailing list has one moderator who receives all the messages that need to be checked. If the moderator decides to accept the message, they simply forward it to the mailing list. Although the moderator may forward the message using their mail client, this will usually give the appearance that they are the author. Mailtraq provides two facilities for moderation without altering the message. You can use the ‘moderate’ button in the Web Administrator (while viewing the moderator’s mailbox), or you can simply drag the message from the moderator’s mailbox to the mailing list in the Mailtraq Console.

Whenever you drag a message in the Mailtraq Console, the message’s *Sender:* field is set to the source mail slot. Since Mailtraq identifies the subscriber using the *Sender:* field in preference to the *From:* field, mailing lists will accept messages moderated in this way.

To assist in moderation across different domains, each message that needs to be moderated will have the header field *X-Moderate-For:* added, with

the full address of the mailing list as the value. When the message is sent back to the mailing list with the *Sender:* field set to the list moderator, the message will be accepted.

- *Note:* you should not disclose the address of the moderator, as it is relatively easy for users with this information to post unmoderated messages to the list. When a message is distributed by the list, the *Sender:* field is set to the address of the list, thus hiding the identity of the moderator.

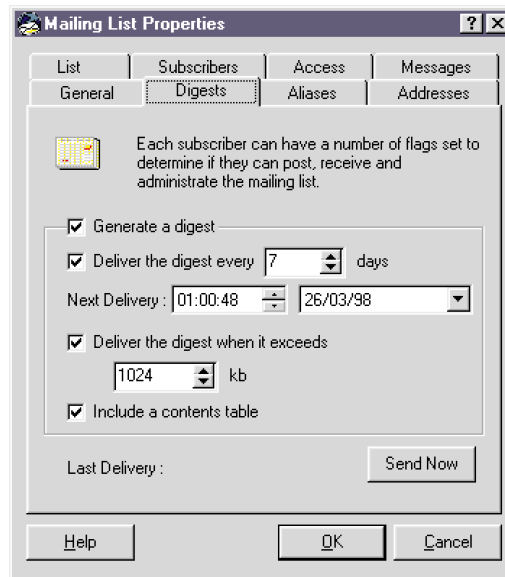
Digests

Subscribers may choose to receive their mail either as an ordinary message (a copy of the message posted to the list) or in the form of a periodic digest. A digest is a collection of several messages in the body of a single e-mail. Mailtraq provides three formats of digest to suit the variety of mail clients in use. The *plain text* option simply appends the message bodies together. Unfortunately this will result in the loss of any attachments, and often makes it difficult to distinguish the individual messages.

Using the MIME¹ standard, the other two digest formats will maintain the individual messages. Most mail clients will show these as separate sub-messages. The difference between these two formats is the MIME type used, and your choice will depend on which type is best supported by the mail client. The default is *Multipart/Mixed*, which has the widest support.

¹ 'Multi-purpose Internet Message Enhancements' is a standard for including data other than plain text in an e-mail message. This allows structural information to be maintained when sent in a message.

*Digest Options
(Mailing List
Properties)*



When digests are enabled for a mailing list, each message distributed by the list is also stored in the mail slot. When the volume of mail in the slot reaches the specified limit, or when the digest delivery time arrives, the digest will consist of the current mail slot contents.

Subscription by Mail

Essentially, this facility allows a user to join and leave a mailing list by sending an e-mail message to a certain address. There are two types of mailing list that can benefit from this feature:

- Discussion Forums
- Announcements Lists

The only real difference between these two types of list is whether new subscribers can post messages. In a discussion forum, everyone should be able to post to the group. With an announcements list, the subscribers expect only the administrators to post messages.

Lists such as these can be greatly enhanced with subscription by mail, from an administrative point of view, and for enticing users to join. Many people who are considering joining such a list may hesitate at the idea of having to be ‘approved’ or explicitly added by the managers. There is a degree of ‘anonymity’ associated with subscription by mail, which many people will appreciate. Another advantage is that users will feel that they can just as easily leave the list if they wish.

Subscription by mail is handled through the sending of ‘control’ messages, which are automatically processed. In Mailtraq, you can send such control messages either to the *mailserver* slot, or directly to the list address. Control messages consist of the ‘command’, followed by any necessary parameters.

Control Messages to *mailserver*

Control messages sent to mailserver must be in a certain format. The command must appear either on the subject line, or as a line in the message body. The command must be the first word on the line, and must be immediately followed by the name of the mailing list to which it is referring. For example, if your mailing list is *new-products@enderson.com*, then the name of the list is ‘new-products’. To subscribe to this list, you might send ‘subscribe new-products’ as a single line in the body of a message to ‘mailserver@enderson.com’.

You can have any number of commands in a single control message to mailserver, providing each is on a separate line in the message body.

Control Messages to the List

You can also send the control messages to the mailing list itself, but this is not as versatile as sending them to *mailserver*. You can only place the command in the subject line of the message, and (with the exception of ‘subscribe’ and ‘unsubscribe’) the command must be prefixed with an ‘*’. This identifies the command from a normal message. Naturally, you don’t need to specify the name of the mailing list when doing this.

Subscribing

To subscribe to a mailing list, you simply need to post a ‘subscribe’ control message as described above. In order to subscribe other people to a list, you will need to append their details to the command. This is called ‘subscription by proxy’, and must be enabled in the mailing list properties.

For example, sending the command ‘subscribe new-products john@enderson.com’ to *mailserver* will add *john@enderson.com* to the new-products list. When you send a normal subscription message, Mailtraq extracts your subscription details from the message headers. The sending address is set first to the *Sender:* field (if present), or alternatively, the *From:* field. The *Reply-To* address is taken (logically) from the *Reply-To:* field (if present), or alternatively, the *From:* field. The subscribers real name is also taken from the *From:* field.

When subscribing by proxy, it is not possible to extract this information from the message, so it can be given explicitly. Mailtraq will assume that the first parameter is the *Sending* address. If another parameter is given (and it looks like an e-mail address), it will be interpreted as the *Reply-To* address. If a parameter is provided within quotation marks, it will be interpreted as the subscriber’s name.

Here is an example command sent to *mailserver* :—

```
subscribe new-products john@enderson.com john-lists@enderson.com "John Smith"
```

Here is the same command, sent in the subject line to *new-products* :—

```
*subscribe john@enderson.com john-lists@enderson.com "John Smith"
```

Confirming Subscriptions

Occasionally users will subscribe to mailing lists without providing a correct return address. This may be due to an error on their part, or perhaps a malicious attack on your system. By confirming subscriptions, each subscriber's address can be verified.

The confirmation process works by returning a message describing the confirmation process to the new subscriber. The message subject line is set to the subscription command, but with an authenticator attached. The message asks the recipient to simply reply if they wish to join the list. When the reply is received, the command will be accepted only if a valid authenticator string is attached.

Secure Lists

If subscription details must be protected, you may enable 'password authentication'. With this facility, every subscriber must verify themselves by appending their password to the subject line of any message they send to the list, and to the end of every command sent to *mailserver*. The password is removed from messages that are distributed.

Subscription Commands

The subscription commands can be used by the subscribers, to alter their subscription account. They can also be used by administrators as a means of remote management. Most of the commands apply to a subscription,

and unless that account is specified explicitly, it will be identified from the control message in which the command appears.

Subscriptions are identified by their Sender address, and consequently no two subscriptions may share that address. When subscription by proxy is enabled, the commands may be applied to a subscription other than the sender by giving the subscription identifier as the first parameter.

The commands available are listed below. For simplicity, the ‘direct to list’ form is shown. For commands sent to mailserver, you would need to add the name of the list immediately after the command name.

Subscribe

There are two forms of this command: the normal form, whereby the sender is the applicant, and the ‘subscription by proxy’ form where another party is to be added.

Subscribe

Subscribe nomail digest

Subscribe “John Smith” john@enderson.com john-lists@enderson.com

The *first* example is the most basic form the command can take. The sender is simply subscribed to the mailing list.

The *second* example also subscribes the sender to the list, but alters their initial flags to those shown. The flags are described along with the set command later.

The *third* example shows the subscription-by-proxy form of the command, where the subscription is created from the parameters. The only required parameter is the subscriber’s e-mail address. You may optionally include

the name of the subscriber, but it must be enclosed within quotation marks. If you include a second e-mail address, it is taken as the *Reply-To* address.

Unsubscribe This command simply removes a subscriber from the mailing list. As with the subscribe command, it can apply either to the sender's subscription, or the subscription specified.

Unsubscribe

Unsubscribe john@enderson.com

The above two examples show the normal and 'subscription by proxy' form of the Unsubscribe command.

Status The status command returns a message describing all the information available on the subscription. This information would include the current subscription settings. If a subscription is not specified, then the sender's subscription status is returned. Only an administrator can request the status of another subscriber's account.

Set This command alters the subscription settings for the sender's account (or another account if specified). The command is followed by one or more flags or settings.

Command	Description
Set mail	Enable ordinary message receipt
Set nomail	Disable ordinary message receipt
Set digest	Enable receipt of the periodic digest, in the MIME format
Set digesttext	Enable receipt of the periodic digest, in the plain text format
Set recbin	Indicates that the subscriber is prepared to accept messages with MIME binary (non textual) attachments
Set norecbin	Disables receipt of binary attachments
Set public	Permits the subscriber's name to appear in the members list
Set nopublic	
Set conceal	Hides the sender's identity from posts to the mailing lists
Set noconceal	Disables the effect of the conceal command
Set name="John Smith"	Changes the name assigned to the subscription
Set password=jmith	Changes the subscriber's mailing list password
Set rec=john-rec@enderson.com	Changes the Reply-To address for the subscription

Suspend and Resume These commands are used to temporarily disable and re-enable the receipt of mail or digests. Subscribers should be encouraged to use this command when they will be away from the list for a long period (to prevent the build up of unread mail for them).

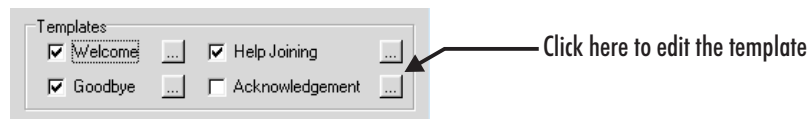
Members This command requests a list of all the subscriptions with the public flag set. Normally this would only consist of administrators, but can also be used for general contacts lists. You should be cautious when providing

public access to this information, as the details may be considered personal and subject to data protection requirements. Further, this presents an ideal list for the sending of unsolicited (junk) mail. For these reasons, subscribers should not set their public flag on lists available outside their organisation.

Message Templates

Often Mailtraq will need to create and send a message of its own. For example, if a message cannot be successfully delivered, Mailtraq will notify the sender with a delivery report. In order for the administrators to have some control over these messages, Mailtraq employs the concept of templates. A template is a base with which Mailtraq can build its own messages, and these can be easily modified.

*Templates used by the
Mailing Lists (Mailing
List Properties)*



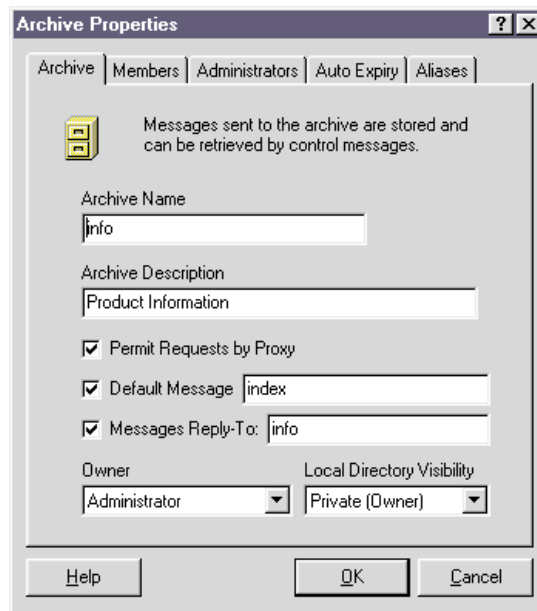
Mailtraq uses a number of templates in mailing lists. When people join a mailing list through subscription by mail, they are sent the *Welcome* template. When they leave the list, they are sent the *Goodbye* template. The *Help Joining* template is used whenever a user writes to the mailing list without having permission to post, and the *Acknowledgment* template is delivered to the sender whenever a message they have posted to the mailing list is distributed. Each of these templates can be enabled or disabled in the mailing list properties.

Templates are discussed in detail in Chapter 16.

Public Archives

Archives have a variety of applications in Mailtraq. Essentially, they provide public access to the messages stored within them. Instead of obtaining the messages through POP3, users request messages stored in the archive by sending a control message. 'Control messages' are handled in a similar way to the mailing lists. They can be sent either to *mailserver*, or directly to the archive. Messages that are clearly not 'control messages', are submitted to the archive for retrieval by others.

Archive Properties
dialogue



It may not be appropriate to allow anybody to add messages, especially since an incorrectly formatted control message may be submitted to the archive. Archives can maintain a list of users who are permitted to retrieve messages (Members) and users permitted to submit messages

(Administrators). If these lists are disabled, then all users are considered *Members* and *Administrators*.

The message sender is identified first by the *Sender:* header field, or if that does not exist, the *From:* field. Because of this, messages distributed by a mailing list take on the identity of that list. This is useful if you wish to have a mailing list automatically add messages to an archive (as mailing lists place their address in the *Sender:* field).

If Mailtraq cannot determine if a given message is a ‘control message’, it relies on the table below :—

Sender is a Member	Sender is an Administrator	Action Taken
No	No	No action taken
Yes	No	Matching messages are retrieved and returned to the sender
No	Yes	Message is submitted
Yes	Yes	Message is submitted

Retrieving Messages

Each message in an archive is automatically given a filename when it is added. Unless specified otherwise, the filename will be the next available serial number. Messages are retrieved by sending a control message specifying the filenames. As with mailing lists, control messages can be sent either directly to the archive, or to the *mailserver* object. Control messages sent to the *mailserver* can have more than one command, while control messages sent directly to the archive have the command in the subject line.

To retrieve a list of the messages in an archive, simply use the *index* command. The message returned will consist of a list of the messages and their corresponding filenames. The filenames listed should be used to retrieve the specified message.

There are three ways to retrieve messages from an archive :—

- *Firstly*, you can use the *send* command, followed by a list of serial numbers. In this case, each message is returned to the sender separately. If you would like to request several messages, they can be returned in a single digest with the *digest* command. You cannot retrieve templates in this way.
- *Secondly* as with mailing lists, commands sent directly to the archive must be preceded with an ‘*’ to distinctly identify them from submissions.
- The *third* option is the implicit *send* command. If the sender is a member, but not an administrator, then the only thing that can do is request messages from the archive. Under these conditions, a user can request articles by sending a control message directly to the archive, with the subject line specifying only the serial numbers. In other words, the *send* command can be omitted.

Messages are returned to the sender of the command. The return address is actually taken from the control message’s *Reply-To:* header, if it is available. Alternatively, the return address will be taken from the *From:* header.

Submitting Messages

Provided either the archive has no limited administration, or the sender is an administrator, they may submit messages to the archive. Under these conditions, any message with a subject line that does not begin with an ‘*’ will be added to the archive.

If you wish to add the message and assign a filename, simply begin the subject line with “*ADD=*filename*”. Filenames must be just a single word.

Templates can also be submitted to an archive. Since Mailtraq cannot automatically identify a template, you will need to begin the subject line with “*TEMPLATE=*filename*”. With both the *ADD and *TEMPLATE commands, the command part of the subject line will be removed before the message is added.

Examples

Here are some examples of control messages sent to *mailserver* to retrieve messages from archives.

Example Archive commands sent to *mailserver*

Index development	This requests a list of all the messages stored in the development archive.
Send help-archive 1, 3-6, 9, 18	This requests messages 1, 3, 4, 5, 6, 9 and 18 from the archive to be returned as individual messages.
Send prices widget	This requests the named file ‘widget’ from the ‘prices’ archive to be returned. This might be a normal message, or a template that generates the prices on demand.
Digest sales 1-99	This requests a digest consisting of all the messages numbered 1 through 99 to be returned. If any of these messages are templates, they will be excluded.

Here are some examples of messages sent directly to the archive. The subject line is shown here.

Example Archive commands sent directly to the Archive

*index	This requests a list of all the messages in the archive.
5	Provided the sender is a member, and not an administrator, message number 5 is returned.
*send 5	Even if the sender is an administrator, the message number 5 is returned.
*add 1998 Price List	Submits a new message to the archive (with the subject line reading "1998 Price List"), and assigns a new serial number as the filename.
*add=Prices Price List	Submits a new message to the archive (with the subject line reading "Price List"), and assigns the filename "Prices" to the message. It can subsequently be retrieved by referring to that name. If a message already exists with that filename, this will replace it.
*digest 5-9	This command requests a single digest consisting of all the messages 5 through 9.
*template=Prices Price List	This adds a new template to the archive, with the template itself consisting of the submitted message body. The subject line is set to "Price List", which is how the template will appear in the index. The filename "Prices" is assigned to the template.

12

The World Wide Web

This chapter discusses the Web related features provided by Mailtraq, including the Web Proxy and the Intranet services. The following topics are covered :—

- What is the Web?
- The Proxy Server
- Web Crawlers
- The Web Server
- CGI and Templates

Although these facilities have little in common with messaging, by having Mailtraq provide them you can avoid separate specialised software, and have tighter integration with Mailtraq's messaging services.

What is the Web?

It is quite possibly thanks to the World Wide Web that the Internet is as popular as it is today. Although e-mail allows people to communicate easily, and the USENET provided a medium for worldwide discussion, there was no way to use the Internet as an effective information resource. The concept of an information store had been around for a long time, in the form of gopher and FTP sites. The disadvantages, though, were that obtaining information from these sources was complex, and finding information was even more difficult.

Information stored in these resources represented a huge variety of formats, some only available to certain users. The result was, as always, authors resorting to the lowest common denominator: plain text. The result was unstructured and uninspiring presentation of information.

HyperText Markup Language

The solution was not to improve accessibility to these resources, but to standardise the way information is stored and presented to the user. Work was already being done in the field of large-scale information management, and a new and very successful standard was emerging: SGML (Standard Generalised Markup Language). A quick and simplified implementation of SGML was quickly developed, called HTML (HyperText Markup Language), and this was to become the standard for the presentation of information.

Unfortunately, HTML was a quick solution, and did not take the usual standards review process. The result was a very good content-oriented markup language, but a poor format-oriented markup language. This means that the markup concentrates on describing the document

structure (such as identifying text as being a heading, or a list, or a paragraph, etc.). Most would (correctly) argue that content-oriented documents are far more important. However, it is very limiting for designers and typographers, who are used to having far more control over the presentation of their documents. The result has been long battle to overcome these limitations, and so many proposed extensions to HTML have been passed about that it is difficult to keep up. The down side is that the appearance of a page can vary considerably from browser to browser.

HyperText Transport Protocol

A new mechanism for transferring this information was also necessary, so a new Internet protocol was carefully developed, and the result was HTTP (HyperText Transfer Protocol). This protocol is both extremely simple, and extremely versatile. The word HyperText in the name might be misleading; the protocol doesn't inherently have anything to do with hypertext. The purpose of HTTP is simply to transfer files between two machines, and at the same time provide reasonable control over the file's format. Another aspect of HTTP is the way files are identified. The URL (Universal Resource Locator) is a way to uniquely identify a file anywhere on the Internet, and HTTP is built around a request for a URL, and a response to that request.

Universal Resource Locators

The URL identifies a specific resource anywhere on the Internet, using a standard representation. The illustration below shows how the URL is constructed.

URL Breakdown



HTML pages usually consist of many URLs, in various contexts. For example, when you click on a hyperlink, the browser loads the page referred to by the hyperlink's URL. When a graphic appears in a web page, it is actually a URL reference to another file (possibly elsewhere on the Internet), which the browser loads and displays on the page.

The Web Now

The World Wide Web is the collection of the concepts described above, and the software used to provide an interface to them. The ability to easily provide information and sites on the Web has led to its rapid growth. However, the concepts behind the World Wide Web are not limited to the Internet. The more recent Intranet concepts have extended this technology towards local networks, taking advantage of the high-quality client software already available for accessing information.

The Proxy Server

A proxy server allows one machine to connect to another through a third party. Normally, such servers are used to provide machines access to the Internet without actually having their own connection to a service provider. Mailtraq can act as a proxy for web browsers, and many other types of service.

The Web Proxy

Most Web Browsers can be configured to use a web proxy. Using a proxy means that instead of connecting to the host specified in the URL, it connects to the proxy. It is then the proxy's task to make the connection and return the requested resource. Only the proxy needs to have a connection to the Internet, and any number of clients can use the proxy services to share that connection.

Each Mailtraq user can separately be given permission to use the proxy facility. To provide access to the proxy for everyone, simply give the default user permission. If the default user is not given proxy access, then Mailtraq will ask for a name and password when the user first uses the proxy. Mailtraq will cache the response, and any subsequent requests from the same machine will be permitted. If the user fails to use the cache for a long period (specified in the Web Server properties as 'Security Timeout'), Mailtraq will ask again for the name and password.

The Web Cache

Proxies can also offer other facilities, such as caching. If several clients are requesting the same web page, then the proxy need only download the page requested by the first client, supplying a copy from the cache for the subsequent requests.

Mailtraq extends this concept to allow the pages to be accessed even when the server is no longer connected to the Internet. This means that you can browse pages of interest, and if you return later, your browser will be able to download the pages without initiating a connection.

Internet Dialup

Mailtraq can initiate a dialup connection if the requested resource is not cached. Either the connection can be made automatically, or Mailtraq can issue a web page asking the user if they wish to connect or not.

Using the Web Proxy

Any web browser on the local network can access the web via the web proxy service provided by Mailtraq. There are three aspects that must be configured on the network :—

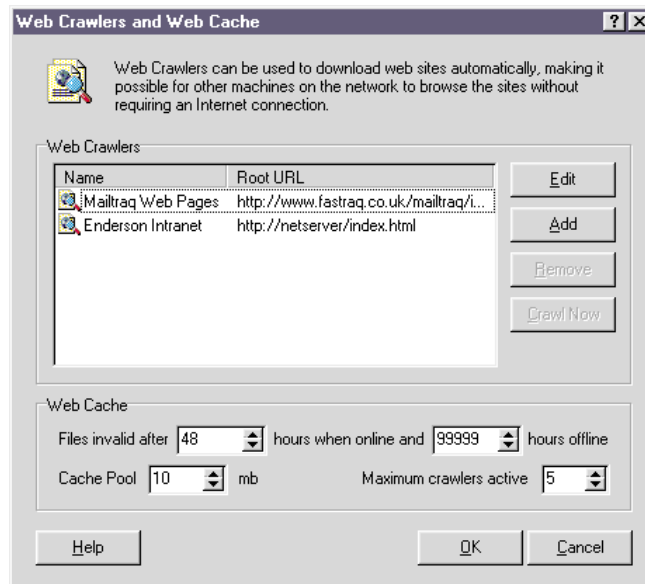
1. The HTTP Service must be enabled and running on Mailtraq, with the Web Proxy enabled.
2. The client machine (from which the web browser is run) must have TCP/IP networking ability.
3. The client browser must be configured to use the Mailtraq web proxy.

Configuring the web browser to use the proxy will vary from browser to browser, and not all web browsers can use a proxy. The browser will need to be given the name of the machine running Mailtraq, and the port on which the HTTP Server is operating.

Web Crawlers

Mailtraq is also able to insert files directly into the web cache using Web Crawlers. Crawlers (to use the common Internet name) are clients that browse web sites in the same way that a user would, but much faster. A crawler will download a page, identify all the hyperlinks, and then follow them.

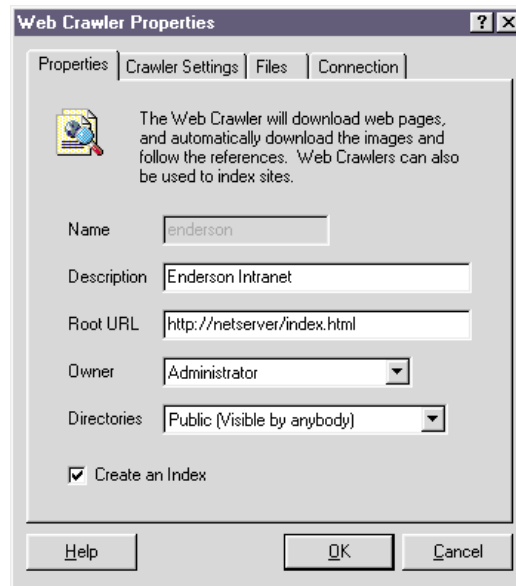
Web Crawlers
Properties dialogue



Every time a file is downloaded, it is added to the web cache. This makes it possible to browse entire web sites without having to actually follow each link by hand while online. Another advantage of the web crawler approach is that sites can be regularly traversed (with only modified files being downloaded), making it possible to mirror a web site. For example, if you may wish to mirror a popular news site for employees to read.

Web crawlers can also index sites. As Mailtraq downloads a web page, it keeps a list of all the words and stores them locally. If you wish to find a web page on a site indexed with a web crawler, you can use the Web Administrator (discussed shortly) to identify pages containing the specified keywords. As Mailtraq is also a web server, you can use this mechanism to index local sites too.

*Web Crawler Properties
dialogue*



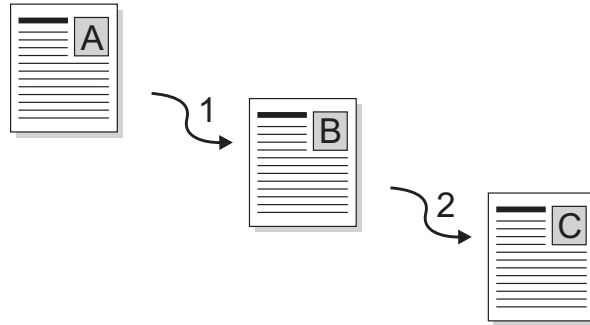
It is important to carefully configure a web crawler in order to make it effective. If a crawler follows every link, it will soon leave the web site and try to download the entire Internet. Further, since crawlers can follow every link, you may find that they download unnecessary files (such as large animations or applications). You will need to set limits on what links it can follow, and what types of files it can download.

Mailtraq allows you to enter a wildcard list of the URLs that the crawler will download [<footnote link to page>](#). You can also indicate if you want the crawler to follow links outside the site you have specified, and how many files to download.

The scan depth determines the maximum number of links to follow, to reach any file. For example, if page A points to page B, which in turn points to page C, then C is two links away from A. You could also say

that C is at depth level two. This is an effective way to force the crawler to concentrate on a specific portion of a web site.

*Counting the Scan
Depth for a Web
Crawler*



Setting the directory visibility (as with mail slots) determines which users will see the site in the Web Administrator.

The Web Server

The Web Server (or HTTP Server) provides the web proxy services described above. It also provides access to the Web Administrator, in addition to serving web pages for a local Intranet. What the web server actually does is determined by the requests it receives.

If the web server receives a request for web pages on a site that is not local, then Mailtraq will act as a proxy for that site. Mailtraq considers the request local if the URL refers to a host that matches the domain or local aliases (as specified in the Server Properties). If a request is received for a page that is local, Mailtraq will return the request (if available) from the local web pages.

Accessing the Web Server

The first stage in configuring the Web Server is to make the server accessible to other machines on the network. You will already have given the machine running Mailtraq a local network name as part of the network configuration. You should tell Mailtraq to recognise this name, and you can do this by adding it as an alias in the Server Properties (accessed in the *Options* menu of the Mailtraq Console). This will allow you to access the web server by using the network name in the URL.

The Web Administrator (discussed comprehensively in Chapter 13) is accessed through a special local path: `'/$'`. For example, if the local machine name is `'netserver'`, then it is accessed through `http://netserver/$`.

Web Pages

Local web pages are retrieved from a directory structure available to Mailtraq. In the HTTP Server Properties, you can specify the root directory for the web server. This directory becomes the root for URL requests to the web server. For example, if you set the root directory to be `'c:\web'`, then the URL `http://netserver/products/index.html` will be translated to `c:\web\products\index.html`.

If you wish to spread the web page directories over different locations, you can use the 'aliases' facility. If you have an alias `'information/company'` set to `'c:\company\details'`, then the request `http://netserver/information/company/index.html` becomes `c:\company\details\index.html`. Aliases can also be relative to the root. For example, the same alias may refer to `'company\details'`, so the same URL will refer to `c:\web\company\details.html`.

Home Pages

Each user may have their own home pages, and their directory is specified in their user properties. This allows browsers to refer to home pages

through the URL *http://netserver/~username*. In order to give the user the opportunity to edit their own pages, they should be given network access to the specified directory.

Editing Pages

Either users may edit their home pages by directly modifying the files on disk, or by using the Web Server upload facilities. Mailtraq supports uploading pages with the *PUT* command. Examples of editors that support this facility include Netscape Navigator Gold 3.0 and Netscape Composer.

CGI and Templates

CGI (or Common Gateway Interface) is a standard for communicating information between the web browsers and the web server, so that web pages can be created dynamically. Mailtraq can use CGI, in combination with Mailtraq Templates, to create dynamic web pages. Chapter 16 gives examples (on pages 227, 229, 239, 241, 258 and 260) of such a system.

Mailtraq will treat any file with the extension *.tpl* or *.mtq* as a template, and execute it accordingly. These pages can be retrieved either with the *GET* command, or the *POST* command. To pass CGI parameters with *GET*, they are appended to the URL (also known as URL-Encoded) after the '?' symbol. Here is an example of such a URL :—

`http://netserver/cgi/database.tpl?db=wines&rec=bot12&field=region`

In this example, the Mailtraq template is *database.tpl*, and the parameters *db=wines*, *rec=bot12* and *field=region* are passed in the URL. With more complex parameters, the URL can become difficult to manage. Further,

sensitive information is often more clearly visible to web browsers in the URL (appearing, for example, in the URL history). The *POST* command simply places the parameters in the body of the request, rather than in the URL.

Mailtraq supports both *GET* and *POST* for templates, but only the *URL-Encoded* format of data (which is also the most widely used). When parameters are sent in this way, Mailtraq converts them to variables and when the template is executed, these variables will already be set.

Given the request above, the *database.tpl* template may contain :—

```
Database Result is { DBRead(db, rec, field) }.
```

In this case the parameters *db*, *rec* and *field* have been placed into variables before the template is executed.

- *Note:* the templates will, by default, appear as plain text (not HTML). This is because Mailtraq sets the outgoing message content-type to text/plain. If you wish to generate HTML output, you should change this with the *SetHeader()* function. Specifically, use

```
SelectOutMessage();  
SetHeader("Content-Type", "text/html")
```

Referring to CGI Templates with HTML

There are two methods for referring to templates and passing parameters to them. You can either directly reference a page, and supply parameters, using the *<A>* HTML tag. For example

**, which will call the specified template with a specific set of parameters. This uses the *GET* command.

To permit the user to specify the parameters, you should use HTML forms. With this method, you can select between the *GET* and *POST* types of request. For example,

`<FORM method=POST action=/cgi/database.tpl>`, in combination with the appropriate `<INPUT>` tags, can be used to pass user-set parameters to the template.

Page 229 provides an example of forms, but this subject is not within the scope of this book.

13

The Web Administrator

This chapter introduces the Web Administrator, which provides remote configuration and access to mail and news from any machine on your network using a web browser. This can serve as a management tool for administrators without physical access to the server machine, and as a means to give users control over their mail slots. The following topics are covered :—

- Accessing the Web Administrator
- User Access and Control
- The Local Directory
- Reading Mail
- Configuring Mail Slots and other Administration Issues

Accessing the Web Administrator

Before you can use the Web Administrator, you will need to configure an HTTP Service with the Web Administrator option enabled.

All pages on the Mailtraq Web Server are accessed through a web browser by referring to the network name of the machine running Mailtraq. In earlier examples, the network name *netserver* has been used. In this case, the pages on the web server are usually referred to with *http://netserver/*.

The Web Administrator is referred to using a system page name: \$. To enter the Web Administrator, use the URL *http://netserver/\$*.

- *Note:* Mailtraq must be able to recognise the name of the machine, otherwise it will assume the user wishes to access the Web Proxy service (to reach another machine). To do this, ensure that whatever names that will be used to refer to the Mailtraq Web Server are added to the Domain Aliases (in the *Server Properties* from the *Options* menu in the Mailtraq Console).

The Web Administrator uses a token security model. This means that every hyperlink that appears in the Web Administrator pages includes a token identifying and authenticating the current user. If one of these tokens is not used within a certain period, the 'Security Timeout' configured in the *HTTP Server Properties*, the user will have to enter their password again.

For this reason, one cannot enter a Web Administrator URL (other than the main menu) manually. It may be more convenient to hide the URL in the browser when accessing these pages.

The Web Administrator Main Menu



User Access and Control

Web Administrator Users are managed from the *Options* menu in the Mailtraq Console. You can specify what the user can access in the Web Administrator by checking flags on the *Web Services* tab of the user's properties. Until a user logs in, they are subject to the settings specified by the 'Default' user. This allows you to provide public access to certain news groups and mail slots.

When a user visits the Main Menu (*http://<server>/*\$) the options they will see are determined by the settings for the Default user (also called the 'Guest' user). In order to access further facilities, the user must log in. Once a valid user name and password have been entered, all the hyperlinks in the web pages generated by the Web Administrator will have

tokens identifying the current user. These tokens are only valid for the current session, and each one will expire after a given period. If you do not use the Web Administrator for longer than this period, you will have to enter the user name and password details again.

Logging in to the Web Administrator



The screenshot shows a Netscape browser window titled "Netscape - [Login]". The address bar shows "File Edit View Go Bookmarks Options Directory Window Help". The main content area displays the "Mailtraq Web Administrator (Guest)" interface. At the top, there is a blue header bar with "Mailtraq Web Administrator (Guest)" on the left and a "Main Menu" link on the right. Below this, the heading "Log In" is prominently displayed. A message states: "Please enter your user name and password below. You may be asked to enter this information again if you do not access any information for more than 20 minutes." There are two input fields: "User name" with the text "johns" entered, and "Password" which is empty. Below these fields are two buttons: "Log in" and "Log in as Guest". At the bottom of the page, a small footer reads: "Generated by Mailtraq (0.8.5.66) on Tuesday, January 20, 1998 02:47:43". The browser's status bar at the bottom shows "Document: Done" and a small icon.

The Directory

This facility provides access to the Mailtraq mail slots. If the current user has administration privileges, they will see all the mail slots, but will only be able to read mail in the slots his or her mailboxes. If the current user does not have administration privileges, then they will see their mail slots listed first, followed by all the mail slots that have protected or public access. The other slots are shown so that the user can write messages to them. Users can only modify the properties and read the mail of their own mail slots.

Local Mail Slot
Directory



The table below shows what mail slots are visible to the currently logged in user :—

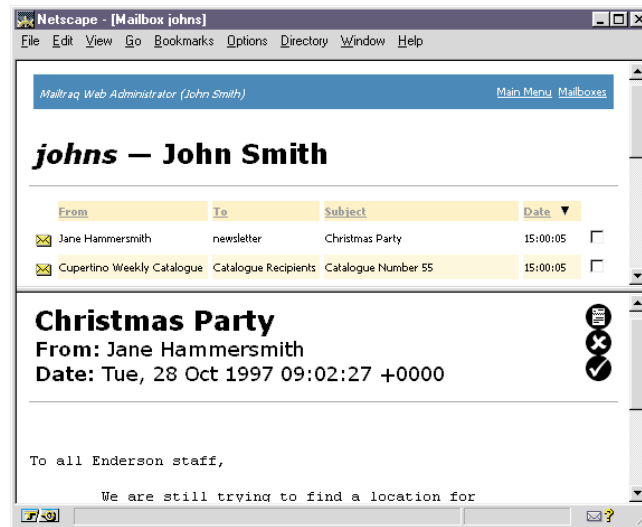
Logged in User	Mail slots owned by user	Other mail slots
Administrator	Read Mail, Properites, Subscriptions	Properties
Normal User	Read Mail, Properties, Subscriptions	Write To... and Subscriptions only if mail slot is Public or Protected
Guest User	n/a	Write To... only if mail slot is Public

If a mailing list appears in a directory, then the current user may use the associated subscriptions icon to subscribe or unsubscribe from it.

Reading Mail


Users can read all the messages in their mail slots by following the Read Mail icon from the directory. The screen is divided into two frames: one shows the list of messages, while the other shows the content of a message when it is selected.

*Reading Mail using the
Web Administrator*



In the above illustration, the message being read was originally sent to the mailing list *newsletter*. Since that mailing list is moderated (by John Smith), he receives the message first. This explains the appearance of the moderate button, which would only appear for un-moderated messages.

*Icons appearing in the
Web Administrator
when reading mail*

-  ← Show Message Headers
-  ← Delete this message
-  ← Accept this message (sent to a moderated mailing list)

The ‘Show Headers’ button on the message viewer toggles whether or not the message headers are visible. If the mail slot has been designated as a moderator for a mailing list, then unmoderated messages will be posted to it. If a message is waiting for moderation, then the ‘Moderate’ button will appear. When you click on this button, the message is posted back to the mailing list (from the moderator mail slot), and will subsequently be accepted. The message is then deleted automatically from the mail slot.

Configuring Mail Slots

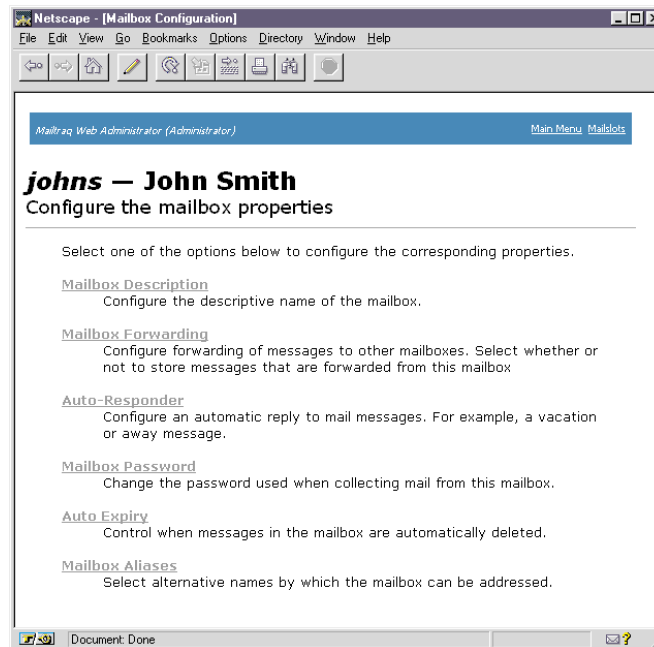
You can configure a mail slot by following the ‘Properties’ links from the directory. When you make changes, you must remember to click on the ‘Apply’ button to actually apply them, otherwise the changes will have no effect. You must also use the ‘Apply’ command before the security timeout expires, or the changes will not be accepted.

Remember that the properties a user may access are determined by their settings in the *User Manager*, available from the *Options* menu in the Mailtraq Console.

Mailbox Properties

The owner’s of mailboxes (and those with System Administration privileges) can modify various ‘day-to-day’ mailbox settings in the Web Administrator.

Configuring the Mailbox Properties

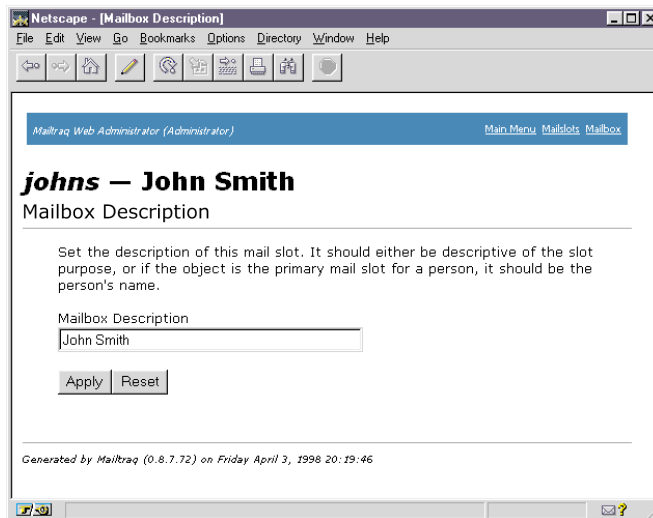


The options that can be set here (other than the Mailbox Aliases) will not influence any other mailbox settings. It is important to note that the Mailbox Aliases (if the user is permitted to alter them) would allow them to capture mail sent to other addresses.

The Mailbox Description

This option allows the user to change the textual description of their mailbox. This only affects how the mailbox is listed in the Web Administrator, the Mailtraq Console, and any public lists issued.

Setting the Mail Slot Description



johns — John Smith
Mailbox Description

Set the description of this mail slot. It should either be descriptive of the slot purpose, or if the object is the primary mail slot for a person, it should be the person's name.

Mailbox Description
John Smith

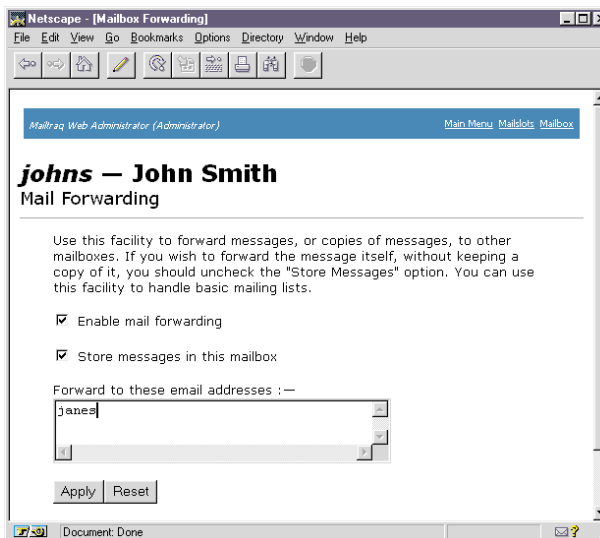
Apply Reset

Generated by Mailraaq (0.8.7.72) on Friday April 3, 1998 20:19:46

Mailbox Forwarding

This option allows users to have mail sent to their mailboxes automatically forwarded to other addresses. Un-check the 'Store in mailbox' option to have the mail passed on to the listed recipients without keeping a copy in the mailbox.

Configuring the Mailbox Forwarding facility



johns — John Smith
Mail Forwarding

Use this facility to forward messages, or copies of messages, to other mailboxes. If you wish to forward the message itself, without keeping a copy of it, you should uncheck the "Store Messages" option. You can use this facility to handle basic mailing lists.

☒ Enable mail forwarding

☒ Store messages in this mailbox

Forward to these email addresses :—
janes

Apply Reset

Document: Done

You may prefer to keep a record of messages forwarded in this way by leaving mail in the mailbox, and simply expiring it automatically instead.

Multiple recipients may be listed — one on each line, either a local address (mailbox name only) or a full e-mail address.

Auto Responder

The auto responder allows replies to be generated automatically in response to incoming messages. This may be a useful way to inform senders that their messages have been received, but not yet dealt with.

Configuring an Auto Responder for a Mailbox

The screenshot shows a Netscape browser window titled "[Mailbox Autoresponder]". The address bar shows "Mailbox Web Administrator (Administrator)". The page has a blue header with "Main Menu", "Mailbox", and "Mailbox" links. The main content area is titled "johns — John Smith Auto Responder". It contains a text area with the following text: "Thank you for your message. I'm out of the office until March 3rd, and I'll get back to you then. If the message is urgent, then please contact Jane Simpson (janes@enderson.com) --- John Smith". Below the text area is a checkbox labeled "Enable the auto responder with the message below" which is checked. At the bottom are "Apply" and "Reset" buttons. A note at the bottom states: "Note: The message can be either normal text, or a message template. Avoid using the '{' and '}' characters in messages that should not be treated as templates."

This is particularly important if you don't plan to respond to the messages (such as receiving answers from a questionnaire), or users may send mail again thinking that their messages have not been delivered.

There are many cases where autoresponders are unnecessary and unwelcome. For example, you should never automatically reply to mailing lists. Many list servers (Mailtraq included) can automatically remove recipients who inappropriately reply to messages. It is also unlikely that you want your automatic response delivered to all the recipients of a discussion forum.

Mailbox Password

The mailbox password refers to that used when authenticating mail clients that collect mail via POP3. Access to the mailboxes through the Web Administrator is authenticated by the user account login.

Changing their Mailbox Password

The screenshot shows a Netscape browser window titled "Netscape - [Mailbox Password]". The address bar shows the URL "http://localhost:8080/mailbox/password.html". The browser's menu bar includes File, Edit, View, Go, Bookmarks, Options, Directory, Window, and Help. The toolbar contains icons for back, forward, home, print, and other standard browser functions. The main content area displays the "Mailtraq Web Administrator (Administrator)" interface. At the top, there is a blue navigation bar with links for "Main Menu", "Mailslots", and "Mailbox". Below this, the user's name "johns — John Smith" is displayed, followed by the title "Mailbox Password". The main text area contains the following instructions: "Please enter a new password and select 'Apply' to make the change. Use this facility to change the mailbox password used when collecting mail (via POP3).". There are two text input fields: "Password:" and "Confirm:". Both fields contain masked text represented by asterisks. Below the input fields are two buttons: "Apply" and "Reset". The status bar at the bottom of the browser window shows "Document: Done".

The password is case sensitive, and will only be accepted if exactly the same text is entered in both edit boxes. You should only use alphanumeric characters (letters and numbers). Other symbols may not

work when typed in from different mail clients, and some characters may be converted by the web browser.

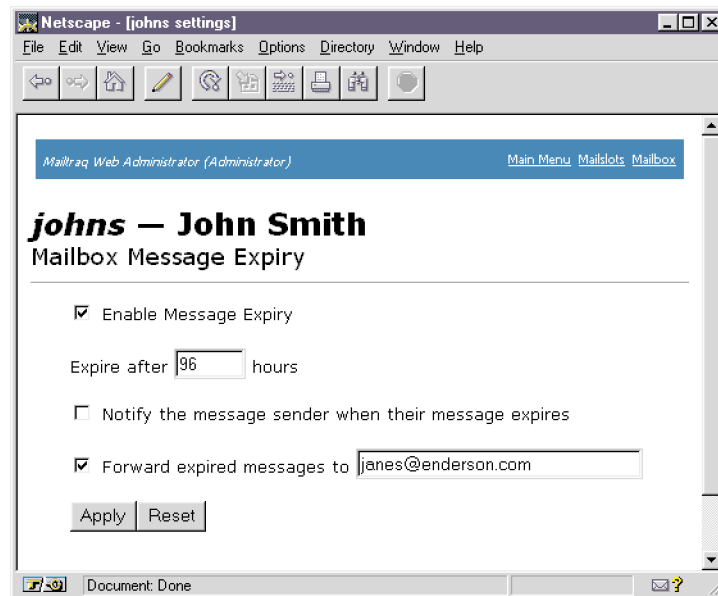
Auto Expiry

This is an effective way to automatically manage mailboxes where mail is not collected by mail clients. For example, a mailbox may be used to keep a copy of all messages sent to a mailing list. In order limit the number of messages stored, they could be expired after a period.

To allow users to manage the auto expire feature of their own mailboxes, this privilege must be granted in the *User Account* properties. This option only applies to mailbox auto expiry, as any user can control the auto expiry feature of their archives.

You may wish to apply an expiry policy of user's mailboxes to ensure that mail is not unnecessarily left in the mailbox, taking up system resources and disk space.

*Configuring the
Message Expiry for a
Mail slot*



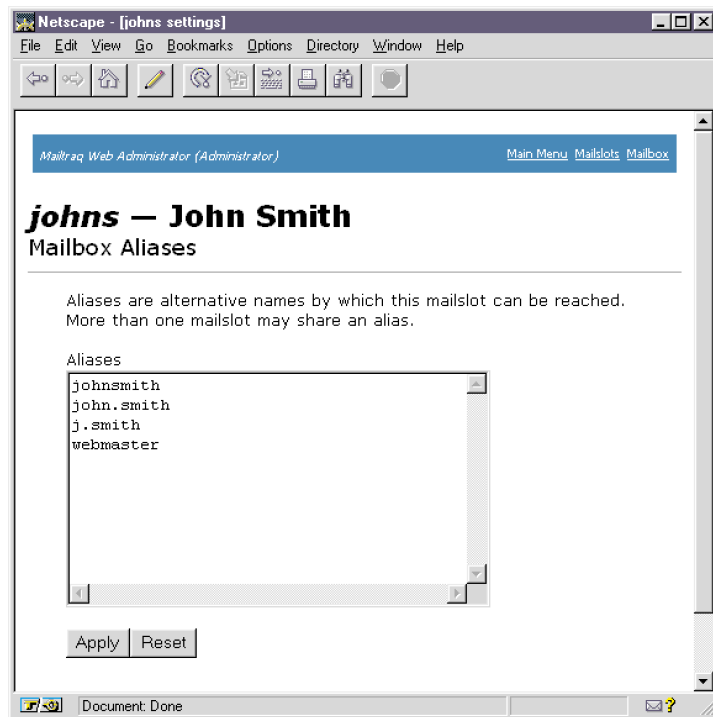
You can also notify users when the messages they sent have been expired, which is an important part of the principle that e-mail never simply be 'lost'.

Having expired messages forwarded to someone else can be a useful way to ensure that the mail sent to certain mailboxes is always read as soon as possible (for example, forwarding mail in an important mailbox to a colleague if it hasn't been read within a certain period).

Mailbox Aliases

This facility allows users to easily configure their mailboxes to receive mail under several identities (without having to centrally control the feature in, for example, the *Address Re-Writing* system).

Setting the Mail Slot
Aliases



As an example, a user may wish to have aliases that include a different spelling of their name to deal with common errors. Another use may be to receive mail under different roles (e.g. “webmaster” if they are responsible for the organisation’s web pages).

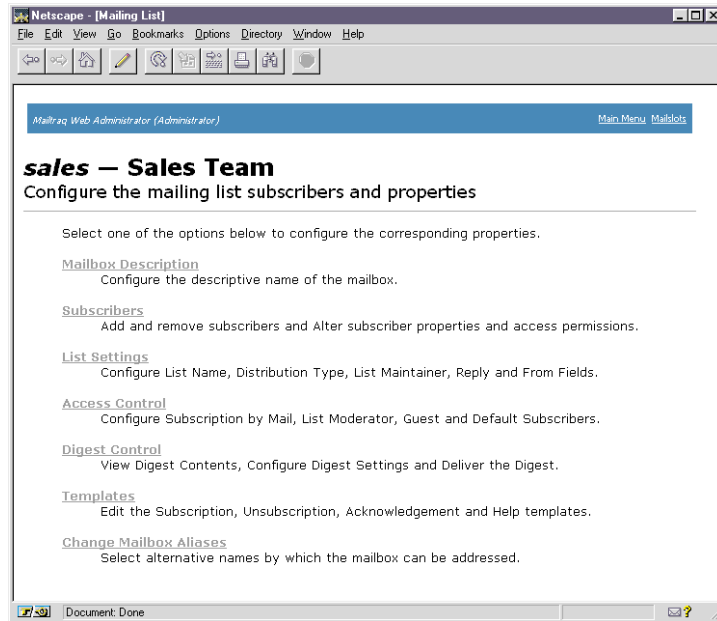
Aliases can also be used to form user groups. Several mailboxes may share an alias, in which case each mailbox will receive a copy of any mail sent to that alias. As an example, several mailboxes may share the ‘sales’ alias, rather than having a central ‘sales’ mailing list.

- It is very important to note, however, that aliases *duplicate* mail — allowing it to be sent both to the named mailbox, and any mailboxes with a matching alias. Users can take advantage of this to receive copies of mail intended for others. If this is likely to be a security problem, the feature should be disabled for non-administrators.

Mailing Lists

Mailing lists can also be configured through the Web Administrator, with many similar options as mailboxes.

The Mailing List Properties Menu



Subscribers

This facility allows you to manage the subscribers of a list, and alter the individual subscription settings.

Managing the Mailing List Subscribers

Mailtraq Web Administrator (Administrator) [Main Menu](#) [Mailbots](#) [Mailing List](#)

sales — Sales Team

Configure the mailing list subscribers

Name ▲	Email	Join Date	Last Post	Delete?
Jane Simpson	janes	28/01/98	28/01/98	<input type="checkbox"/>
John Smith	johns	28/01/98	28/01/98	<input type="checkbox"/>
Kate Pearson	katep	28/01/98	28/01/98	<input type="checkbox"/>
Mark James	markj	28/01/98	28/01/98	<input type="checkbox"/>

Generated by Mailtraq (0.8.7.72) on Saturday April 4, 1998 02:18:43

Mark James

Email:

Reply-To:

Details:

Password:

☐ Suspended
☒ Recipient
☒ Sender
☐ Admin
☐ Moderated
☐ Barred
☐ Digest
☐ Receive Binaries
☒ Send Binaries
☐ Digest multipart/digest
☐ Digest text/plain
☐ Public Listing
☐ Conceal

Document: Done

The left window pane shows the current subscribers. You can click on the column headings to sort according to that column, and if there are more subscribers than can be shown in one page, the page control icons will appear.

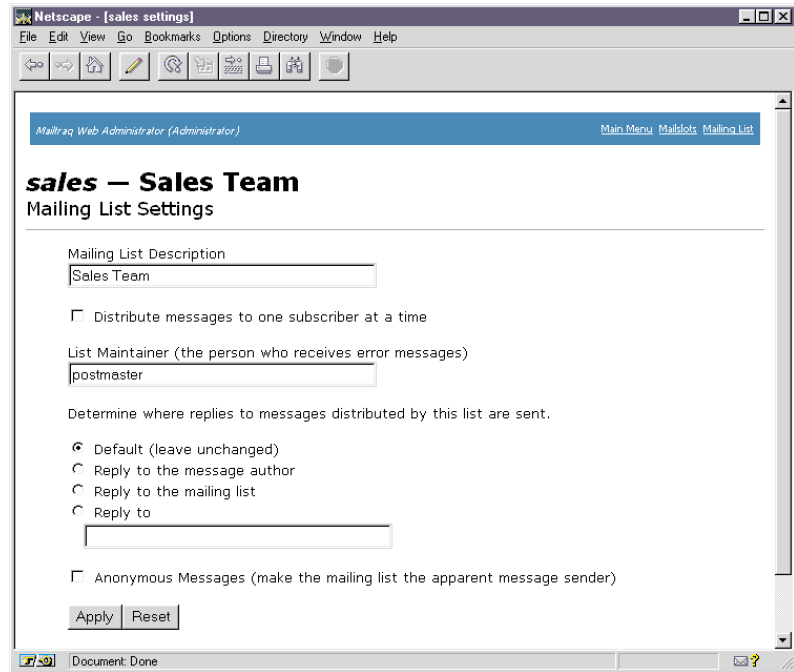
The right window pane shows the details of an individual subscriber. To select a subscriber, click on the icon. To delete one or more subscriber, check the 'Delete' boxes in the rightmost column, and then click on 'Delete Selection'.

When changing the individual subscription settings, don't forget to use the 'Apply' button to save the changes.

List Settings

This facility allows you to configure the general mailing list settings.

Configuring a Mailing
List



Distribution Type

The 'Distribute messages to one subscriber at a time' spreads the volume of incoming mail evenly amongst all the subscribers. An example might be a support list, where Mailtraq selects one subscriber at a time to handle a technical support query. You can always have mailing lists as subscribers in order to spread the mail amongst several groups of people.

List Maintainer

The List Maintainer is essentially the name put in the 'From' field of the message envelope (not to be confused with the message header field), and in the *Errors-To:* header field. These are the recognised methods for directing delivery failures to specific people (as a means of avoiding delivery failures being returned to the mailing list itself).

The maintainer should read these messages and take appropriate action, for example removing subscribers from the list when messages cannot be delivered to them.

Reply Address

The reply address is likely to depend on the nature of the list. For discussion forums, replies should generally be directed to the list itself (so that all the subscribers can read them), whilst support lists should direct replies to the individual authors. In both cases, the setting simply affects the default action taken by the mail client, and it should be possible to override this.

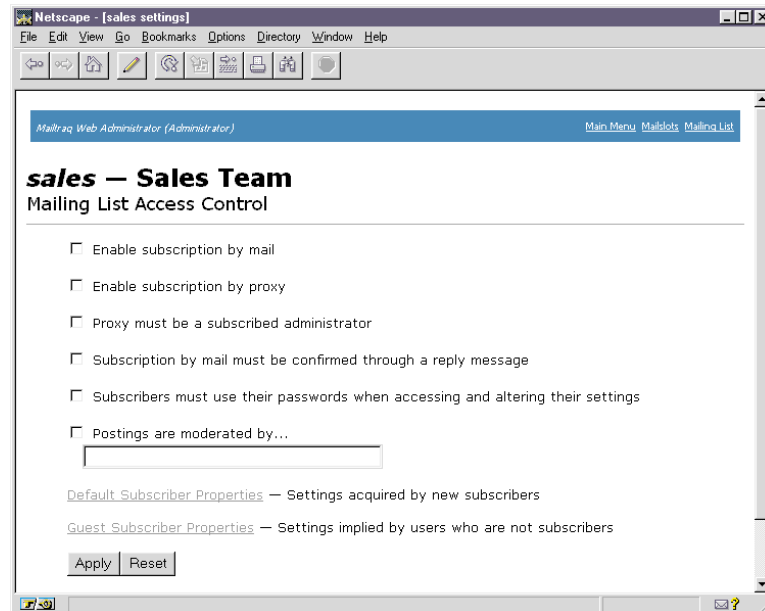
Anonymous Lists

The anonymous list option hides the identity of the original message author, by making the message appear to be created by the list itself.

Access Control

This facility allows you to control how subscribers access the mailing list, and what options and controls are applied to them.

Configuring the Mailing List Access Control



The Subscription by Mail system is described in detail in Chapter 9. However, one important setting here is the ‘Moderated by...’ option.

When message moderation is enabled, subscribers with the ‘moderated’ flag set must have all their messages verified before they can be distributed by the mailing list. Mailtraq provides a facility in the Web Administrator to help with this process.

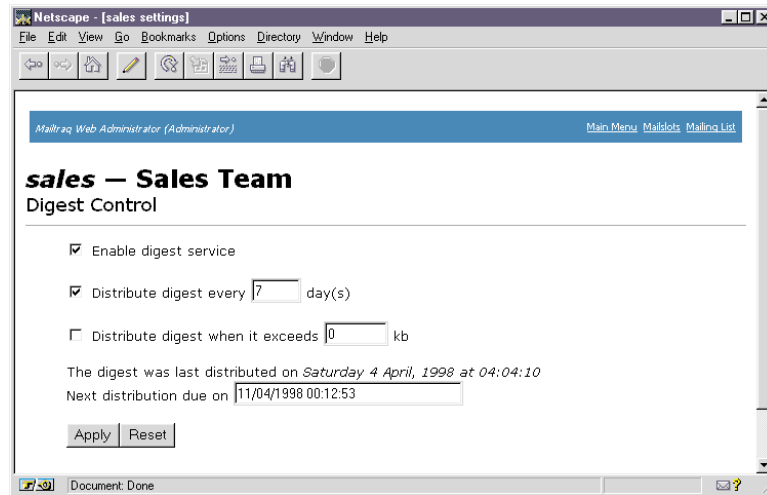
When reading mail from the moderator’s mailbox using the Web Administrator, messages awaiting moderation have additional button which allows the moderator to accept the message, automatically forwarding it to the mailing list and deleting it from the mailbox.

Digest Control

The digest control facility allows you to configure the automatic list digest system. Digests are simply collections of messages posted to the list, in the form of a single compound message. Two of the digest formats supported by Mailtraq are MIME (Multipurpose Internet Mail Extensions), which means that many mail clients will be able to interpret them as message collections.

Digests can be generated periodically, and users who do not need to be kept as up-to-date as other subscribers may prefer this method, as it results in fewer e-mail messages arriving in their mailboxes.

Configuring the Mailing List Digest service



Mailing List Templates

The Templates are messages which are generated automatically in response to various conditions (such as a subscriber joining the list via an e-mail application).

Refer to Chapter 11 for more information on the mailing list templates, and to Chapter 16 for information on the Mailtraq template system.

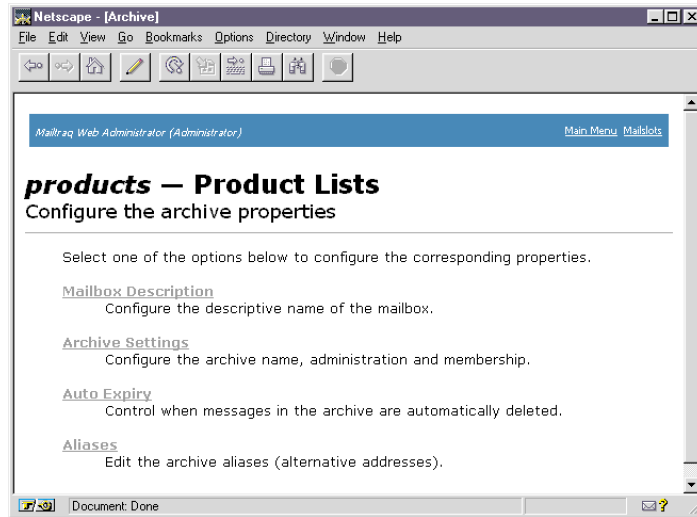
Archives

The Mail Archives are used for providing public access to the messages that they contain. Unlike mailboxes where messages are held for a specific user, who collects them (and thus removes them from the mail slot), the archives allow users to retrieve the messages by request.

Any e-mail message can be stored in the archive, and when a request is made the archive message is returned in the same format as it was received in the archive. Requests are made by sending a control message to the 'mailserver', as described in Chapter 11.

The archive properties available to the Web Administrator are very similar to the mailboxes, and these are shown below.

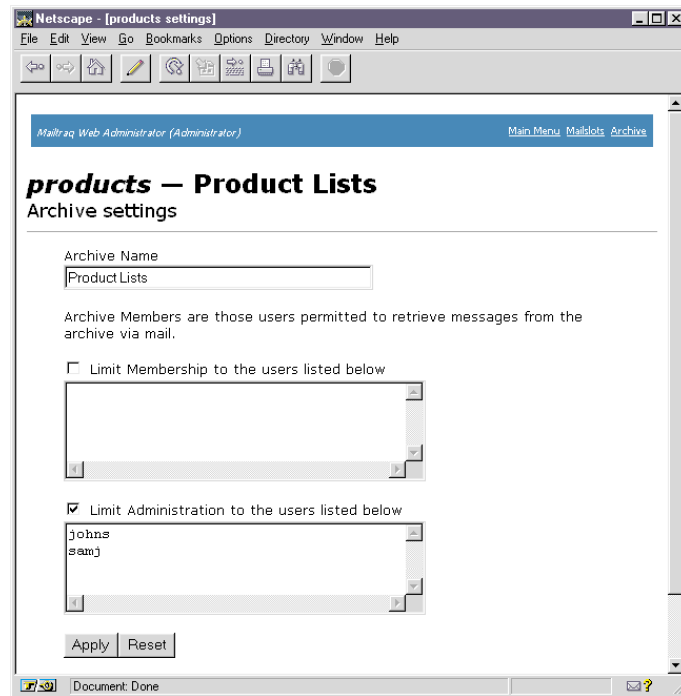
The Archive Properties Menu



Archive Settings

The Archive Settings essentially allow you to configure the ‘members’ and ‘administrators’ for the archive, which control the access that users have to the archive messages.

Configuring the Archive Members and Administrators



Members

The 'members' are the users (described by their e-mail address) who are permitted to request messages from the archive. Mailraaq examines the *Sender:* and then *From:* header fields of the message to determine if the sender can access the archive.

Administrators

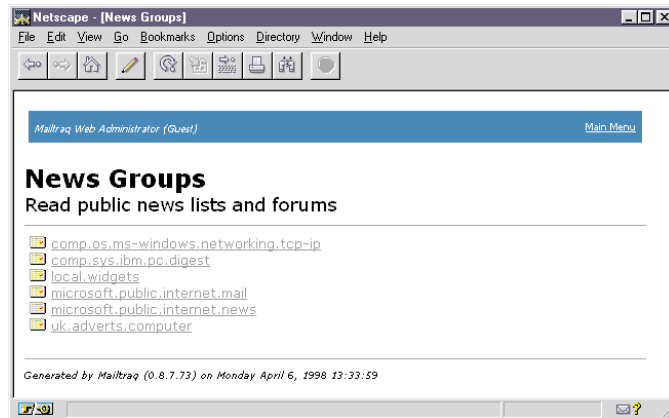
The 'administrators' are those people permitted to add messages to the archive. Again, Mailraaq examines the *Sender:* and (in the absence of that field) the *From:* field. This is done because most mailing lists (and all the Mailraaq mailing lists) set the *Sender:* field to the address of the list itself. By simply adding this address to the 'administrators' list, the archive can then accept any message distributed by that list. By making the archive a receiving subscriber of the mailing list, it can be used as an effective way to archive the list messages for public access.

Reading News

The Web Administrator provides a primitive interface to the news groups stored in the Mailtraq database. It is primitive in the sense that it does not provide many features that you may expect to find in news clients, such as article threading and searching. However, this does provide a simple means to access the news groups without having to install and configure a news client.

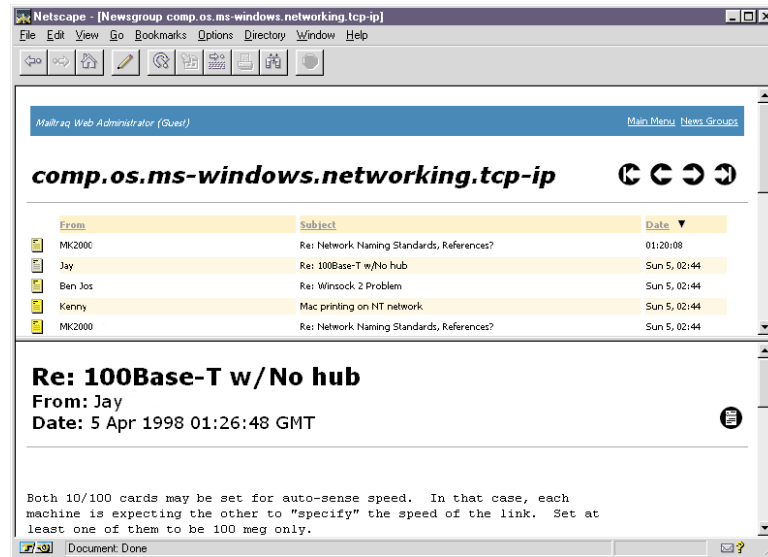
Mailtraq will use the news group access controls specified in the user accounts when offering news groups for reading. An example of such a list is displayed below.

Selecting a News Groups



The articles are shown in the order that they appear in the database. If more articles are available than will fit on the page, the page navigation icons will appear.

Reading News Articles



The top frame shows the list of articles, while the frame below shows the contents of the current article. To select an article to show in the lower frame, click on the article icon (in the article list).

Accessing Web Sites

The 'Web Sites' option allows users to access other web sites through Mailtraq. The list that appears represents the currently configured *Web Crawlers* (accessed through the *Options* menu in the console).

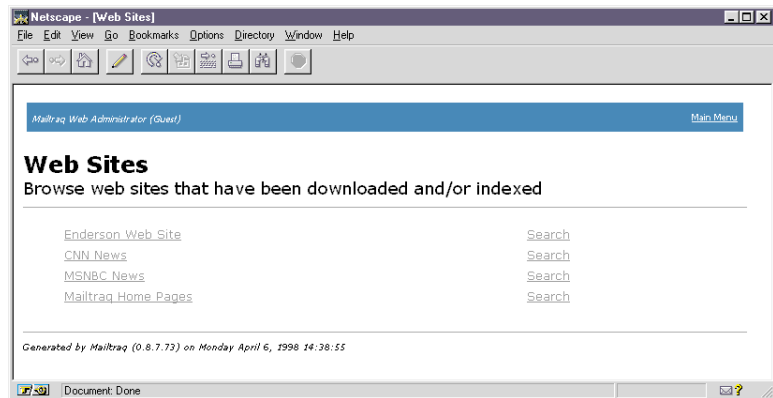
Mailtraq will only display the web sites that have an appropriate 'visibility' configured. For example, until the user has logged in they will only be able to see *Public* web crawlers/sites.

When the user clicks on the link to the listed web site/crawler, the web browser will follow that link to the specified site — which may result in the browser leaving the Web Administrator.

However, if the browser is configured to use Mailtraq as its web-proxy, then it will request that page from Mailtraq, and if the page is held in the cache, Mailtraq will be able to provide it without going on-line. This allows Mailtraq to scan web sites and download the pages in advance, providing access to them through the web-proxy facilities.

If the page is not available, Mailtraq can go on-line — asking the user if configured.

Accessing Web Crawler Sites



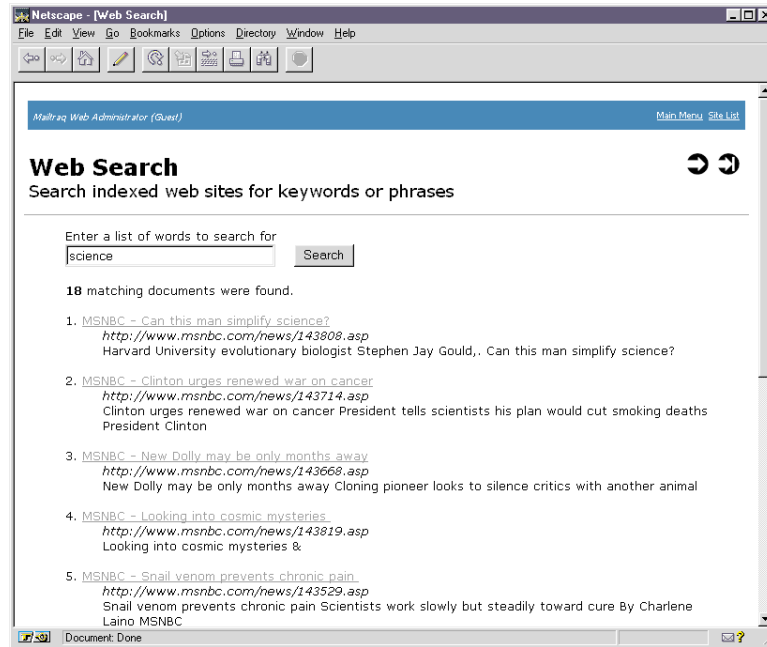
The Web Crawler can also monitor local web sites, such as those provided by the Mailtraq web server. The advantage for this is that the sites (as with any web site) can be indexed, and the user can search for keywords.

Searching Web Sites

When the user clicks on the Search link, they will be asked to enter one or more keywords. Mailtraq will then scan its index of that site for matching keywords, and display links to the pages it has found.

Mailtraq can only provide the searching mechanism for web pages that it has ‘crawled’.

*Searching for keywords
in a Web Site*



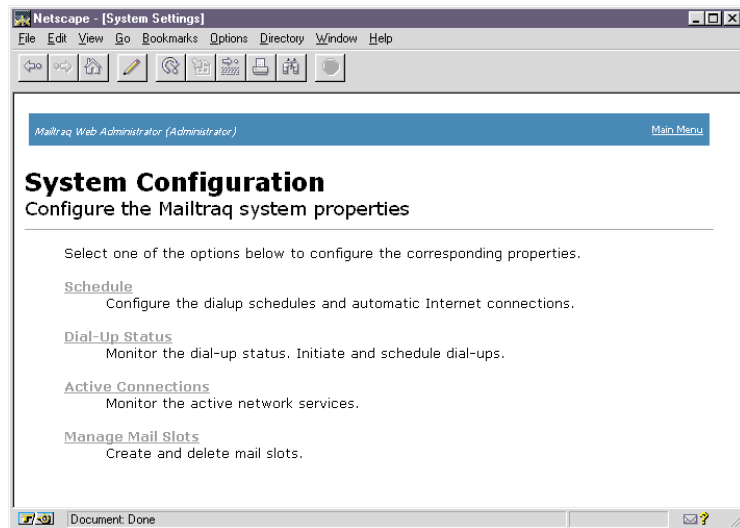
In the example above, the keyword “science” has been entered, and Mailtraq is displaying the pages it has matched. Mailtraq will sort the matching pages according to how often it has found the matching keywords, showing the pages with the most matches first.

System Administration

If the current user has ‘System Administration’ privileges, they will be able to access the options in this section. The System Administration facility

allows users to monitor and configure some of the system parameters, such as network dial-up and the creation of mail slots.

System Configuration
Page



Schedule

This option allows the user to configure the dial-up schedules. A list of the currently configured schedules is shown, and in the frame below the currently selected schedule settings are displayed. To select a schedule, click on the icon beside it.

Configuring the Dial-up Schedules

The screenshot shows a Netscape browser window titled "[Schedules]". The address bar shows "Mailtraq Web Administrator (Administrator)". The page has a blue header with "Main Menu" and "System Settings" links. The main content area is titled "Dialup Schedules".

Active	Description	Delete?
<input checked="" type="checkbox"/>	Daily at 08:00:00 (News) (POP3) to ISP Line #1	<input type="checkbox"/>

Buttons: Apply Add Schedule Reset

Dialup Settings

Current RAS Connection: ISP Line #1

☐ Automatically select next alternative on failure to connect

Number of retries: 3

Disconnect after 120 seconds of inactivity (0 disables)

Disconnect unconditionally after 900 seconds

Schedule Date

☒ Connect Every Day

☐ Connect only on the following days

☐ Sunday ☐ Monday

☐ Tuesday ☐ Wednesday

☐ Thursday ☐ Friday

☐ Saturday

☐ Connect on specific day of month 1

Schedule Time

☒ Once a day at 08:00:00

☐ Every 0 minutes past the hour

☐ Every 480 minutes (starting at 12am)

☐ Only between these times 00:00:00 and 23:59:00

Schedule Options

☐ Enable only if outbound mail waiting

☐ High-priority mail only

☐ Enable only if offline for more than 0 minutes

Buttons: Apply Reset

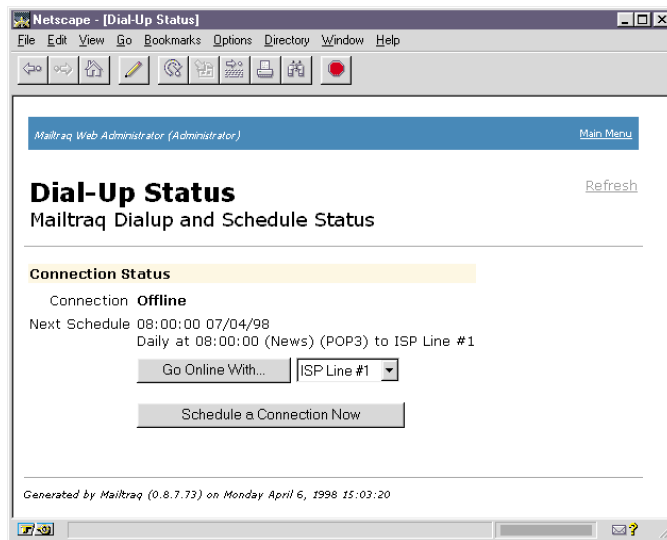
Each schedule also has checkboxes to set whether or not it is active, and to select it for deletion. In both cases, Mailtraq only activates or deletes the schedules when the 'Apply' button is pressed.

The schedule settings themselves are the same as those configured from within the Dial-up properties in the Mailtraq Console.

Dial-Up Status

This option allows the user to monitor the dial-up service, and manually go on-line and off-line. From here automatic schedules can also be initiated. The page also gives details of the next scheduled dial-up.

*Monitoring the
Mailtraq Dial-Up
Status*



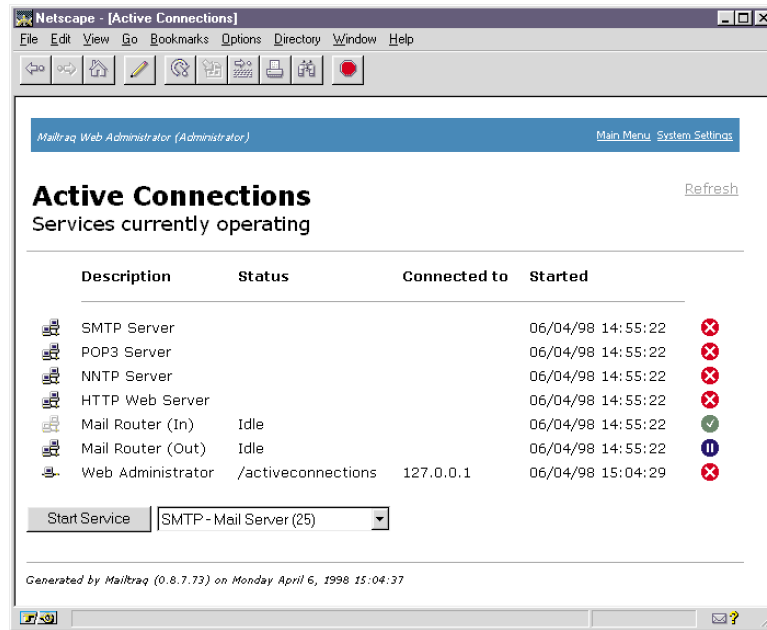
To update the information, click on the 'Refresh' link. If you are using a Netscape browser to view the Web Administrator, Mailtraq will automatically update the page every few seconds.

Active Connections

This facility is very similar to the 'Active Connections' tree in the Mailtraq Console. A list of all the current services are shown, and by clicking on the icon to the right of each connection, you can disable or close them. The two routers can also be suspended and resumed with the icon to their right.

To initiate a service, select it from the drop down list and click on ‘Start Service’. If the service has already been started, no action will be taken.

Monitoring the Active Network Connections



It is important that you have an up-to-date page before you close any of the services, as it is possible to close the wrong service if old information is shown. You can refresh the page with the ‘Refresh’ link.

If you are using a Netscape browser, the page will be refreshed automatically every few seconds. If this prevents you from starting a service, simply press the stop button in your browser.

Manage Mail Slots

This facility allows you to create and delete mail slots from the Mailtraq system. Once a mail slot has been created, a link will appear to allow you to configure its properties.

Creating and Deleting Mail Slots

The screenshot shows a Netscape browser window titled "[Manage Mail Slots]". The address bar shows "Mailtraq Web Administrator (Administrator)". The page has a blue header with "Main Menu" and "System Settings" links. The main content area is titled "Manage Mail Slots" with the subtitle "Create and delete mail slots".

Create a new mail slot

Mail slot name:

Slot Type: Directory Visibility:

Mail Group:

Mail Slot Owner:

Delete an existing mail slot

Mail Slot:

The browser's status bar at the bottom shows "Document: Done".

14

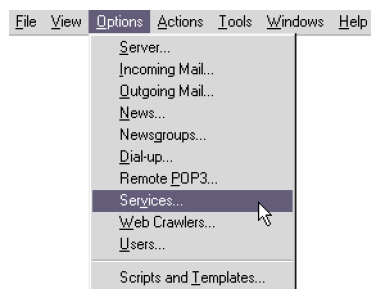
Network Services

This chapter discusses the Network Services, which are the facilities that Mailtraq provides to which other machines (or applications on the same machine) can connect.

The following topics are covered :—

- The Service Manager
- Using Firewalls
- The Mailtraq Services

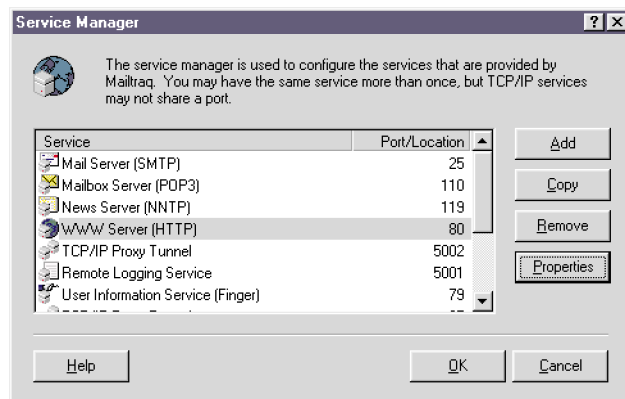
Mailtraq's services are all configured from the *Services* item in the *Options* menu of the Mailtraq Console.



The Service Manager

The Service Manager dialogue allows you to configure and add new networking services, as shown below :—

*The Service Manager
dialogue*



Each services has a 'port' or 'location' where it waits for incoming connections. The TCP/IP services have a numeric port, most of which

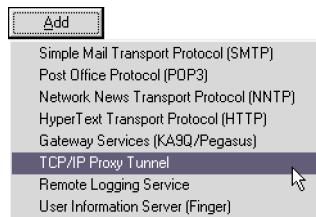
are standardised. For example, applications will expect to find the SMTP service listening on port 25.

No two TCP/IP services may listen on the same port at the same time, but Mailtraq does allow you to *configure* multiple services on the same port. You can load individual services as needed from the *Actions* menu of *Console*, and close them by right-clicking on the service in the *Active Connections* list.

The Mail Gateway is not a TCP/IP service, but instead monitors a directory for new files. When a message is placed in the directory being monitored, Mailtraq reads the file and processes it as an incoming mail message.

Adding a service

To add a new service, click on the 'Add' button and select the type of service from the popup menu that appears.



Modifying Services

When you click on the 'Properties' button, the service's configuration dialogue will appear. The TCP/IP services share many attributes, which will appear in several of the configuration dialogues.

Enable at Startup

If this option is selected, then when Mailtraq is loaded the service will be automatically started. Any service can be started from the *Tools* menu of the *Console*.

Port to listen on Unless you are providing an alternative or proxy service, you should use the standard ports shown below :—

Service	Recommended Port
SMTP	25
POP3	110
NNTP	119
HTTP	80 (or 8080 for the proxy)
Finger	79
Remote Logging	Any port over 5000
Proxy Tunnel	Any port
Mail Gateway	N/A

Firewalls

A firewall is one of the most basic forms of security for services, and is an effective access control method. It is also a simple way to prevent clients from automatically using services, such as preventing the Mail Server from your Internet Service Provider delivering mail via SMTP (when you wish to actively collect via POP3 instead).

The firewall works by listing the IP addresses of the machines you wish to allow access to. The IP address is the unique 32 bit number assigned to every machine on the Internet. (Machines only visible on your local area network are unlikely to be globally unique, as the network administrator may select them.)

IP addresses are represented by four numbers, separated by periods, in the range 0–255. For example, ‘192.168.0.1’.

*Use 192.168.0.1
through 192.168.0.255
for local IP numbers*

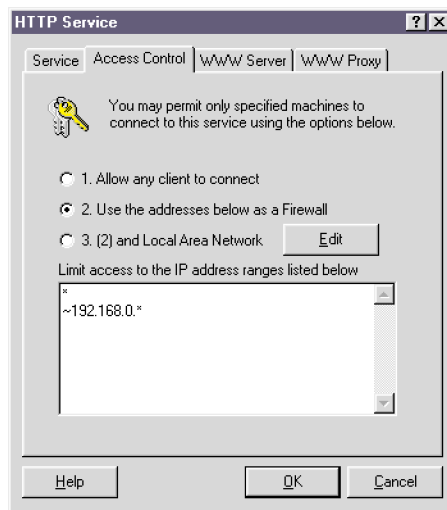
It is recommended that you use the IP addresses ‘192.168.0.1’ through ‘192.168.0.255’ for your local area network, as these numbers are guaranteed not be assigned to any Internet machine.

Mailtraq allows you to specify ranges (such as that above) by using the ‘*’ wildcard. For example, to specify the range above, you could enter ‘192.168.0.*’.

Mailtraq also allows you to exclude a single address, or range of addresses, by prefixing them with a tilde (~). For example, ‘~192.*.*.*’ would prevent any machine with an IP address starting with 192 from connecting.

The ranges are interpreted from top to bottom, and firewalls begin empty (excluding everything) so if you wish to use an exclusive firewall, you should begin with an ‘*’ — otherwise all numbers will be excluded.

*HTTP Service
Properties*



Local Area Network List

All the service firewalls can also refer to the Local Area Network list, which defines the IP addresses of the machines on the local network. These addresses can be used as a firewall that is shared by any of the services. You can configure the Local Area Network list directly from the *Server* properties (accessed from the *Options* menu in the Console).

The Mailtraq Services

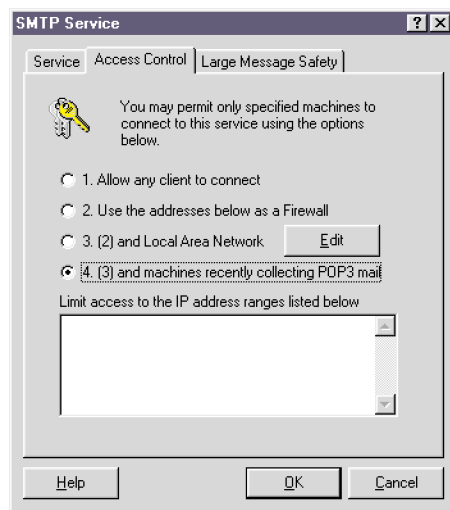
The SMTP Service

The 'Simple Mail Transfer Protocol' allows mail clients to deliver e-mail messages to Mailtraq. It is described in more detail on page 74.

Access Control

The SMTP service offers an additional Access Control option: '...machines recently collecting POP3 mail'. With this option selected, any machine that successfully connects to the POP3 service is noted, and are permitted to connect to the SMTP service if they attempt within a few minutes.

SMTP Service
Properties



Many mail clients first connect to the POP3 service to collect mail, and then begin delivery via SMTP. This method allows those mail clients to gain access to SMTP, effectively using the mailbox authentication mechanism used in POP3.

Large Message Safety

This facility can be used as a defence against unnecessarily large messages. Such messages can tie Mailtraq up on your dial-up connection for a long (and possibly costly) time, and such misuse may also be malicious.

SMTP does not have any means to refuse large messages (as POP3 does), but Mailtraq can cancel the connection and refuse to accept the same message next time a delivery attempt is made (as mail servers should try to send the message again if the connection is cut during delivery).

The POP3 Service

The 'Post Office Protocol' service allows mail clients to collect messages from the mailboxes managed by Mailtraq. POP3 is covered in detail in Chapter 6.

It is unusual for mail clients outside your local network to connect to the POP3 service, particularly if you have a dial-up connection. For this reason, it is usually recommended that you implement a strict firewall on this service (in order to safeguard access to the messages in the local mailboxes).

The NNTP Service

The 'Network News Transfer Protocol' service allows news clients to collect (and post) news articles from the news groups handled by Mailtraq. The Mailtraq news facilities are covered in detail in Chapter 10.

It is possible for Mailtraq to authenticate clients by asking for their username and passwords and matching them against the *User* records (accessed from the *Options* menu in the Console).

The HTTP Service

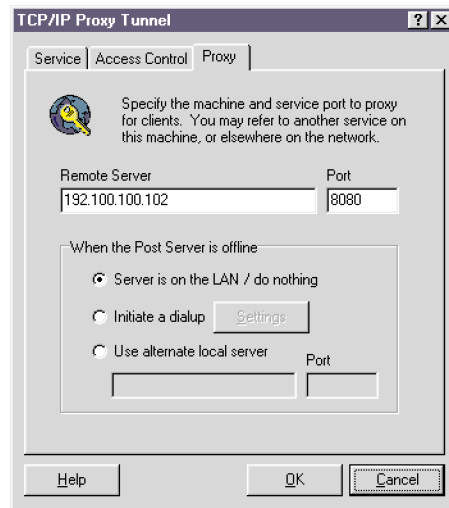
The 'HyperText Transfer Protocol' service provides a number of different functions. It can act as a Web Proxy, a Web Server and as a host for the Web Administrator. These facilities are described in more detail in Chapter 12.

The TCP/IP Proxy Service

This service allows Mailtraq to host a tunnel proxy between a service on another port/server on the Internet and the local network.

When a client connects to this service, Mailtraq will pass the connection on to the service specified in the Proxy tab.

TCP/IP Proxy Tunnel Properties



If the machine specified is not on the local network, you can have Mailtraq start an automatic dialup schedule. Mailtraq can also connect to an alternative service when it is offline.

This service can be used as an effective outbound and inbound proxy. An outbound proxy would be to allow local machines to connect to services on the Internet via Mailtraq. Conversely, an inbound proxy would allow other machines on the Internet to connect to services offered by machines on the local network other than the dial-up machine. This can also be used to redirect connections for services that are on non-standard ports.

For example, if you wish to have inbound mail handled by a machine other than the dial-up host on which Mailtraq is running, you can proxy port 25 to a service on the local network. This example can be extended by having Mailtraq run on several machines, each equipped with dial-up facilities, but where all the incoming SMTP connections are connected via a TCP/IP tunnel proxy to the mail server. Using this system, incoming mail can be received when any machine running Mailtraq is on-line.

This proxy is quite different to the proxy service offered by the HTTP service (which can only be used to proxy HTTP connections). This type of proxy can carry any single stream TCP/IP service (including all the services supported by Mailtraq). It specifically cannot carry FTP connections, because they require two streams.

The Remote Logging Service

This service simply allows remote machines to monitor Mailtraq activity by receiving all the log messages. You can connect to this service using Telnet, which provides a clear text transmission. There is scope for building software to analyse the data in more detail.

The User Information (Finger) Service

This service has historically been used to provide information on users of mainframes. However, it is also an effective way to determine if mail is

waiting for a user, which is what Mailtraq does by default in response to finger requests. The response given is taken from the *finger_server.tpl* template.

The Gateway Service

This is the only non-TCP/IP service. It simply monitors a directory for new files, and imports messages according to the KA9Q and Pegasus mail gateway standards.

When the message is read it is inserted into the Inbound Mail Router and processed before being deleted from the gateway directory.

KA9Q Mail

When Mailtraq detects the arrival of a file with the .txt extension, it assumes that it is in KA9Q format. Messages in this format are stored in a pair of files: one with the .txt extension (containing the message itself) and a file with the .wrk extension identifying the sender and the message recipients (similar to the message envelope).

■ .WRK Files

The .wrk file must have the same filename (before the extension) as the .txt file, and have the following format :—

```
hostname of sender.␣
sender's e-mail address.␣
recipient's e-mail address.␣
further recipient addresses.␣
...
```

This can be described in BNF¹ as :—

```
wrkfile      ::= hostname CRLF sender CRLF recipients
hostname    ::= // hostname of sender
```

```
sender          ::= // e-mail address of sender
recipients      ::= recipient | recipient CRLF recipients
recipient       ::= // e-mail address of recipient
CRLF            ::= // newline
```

One or more recipients may be listed, as shown in these example files :—

Single Recipient Example	Multiple Recipient Example
enderson.com	mail.widgetmakers.com
johns@enderson.com	widgets@widgetmakers.com
janes@enderson.com	johns@enderson.com
	janes@enderson.com

The `.txt` file should contain a complete e-mail message (in RFC822 format) including headers (an example of which is on page 57).

Pegasus Mail Pegasus messages are stored in a single file, with the `.cno` extension. These files include the message envelope in addition to the message itself.

■ .CNO Files

The `.cno` file must have the following format :—

```
F sender's e-mail address
T recipient address
C carbon copy recipient address
B blind carbon copy recipient address
... more recipient lines
message text
```

1 BNF (Backus-Naur Format) is a formal specification language for defining syntax

More than one T, C or B line may be given containing recipient addresses, and the C and B lines are optional.

This can be described in BNF as :—

```

cnofile      ::= sender CRLF recipients CRLF message
sender       ::= "F " email
recipients   ::= recipient | recipient recipients
recipient    ::= "T " email | "C " email | "B " email
email        ::= // e-mail address of sender or recipient
message      ::= // message text including headers

```

Here are some examples of the headers from .cno files :—

Example 1

```

F johns@enderson.com
T janes@enderson.com
C piercet@enderson.com

```

Example 2

```

F johns@enderson.com
T widgets@widgetmakers.com
B johns@enderson.com
B janes@enderson.com

```

- *Note* that the message text (including header fields) follow immediately after the recipient addresses.

More than one T, C or B line may be given containing recipient addresses, and the C and B lines are optional.

This can be described in BNF as :—

```

cnofile      ::= sender CRLF recipients CRLF message
sender       ::= "F " email
recipients   ::= recipient | recipient recipients
recipient    ::= "T " email | "C " email | "B " email
email        ::= // e-mail address of sender or recipient
message      ::= // message text including headers

```

Here are some examples of the headers from .cno files :—

Example 1

```

F johns@enderson.com
T janes@enderson.com
C piercet@enderson.com

```

Example 2

```

F johns@enderson.com
T widgets@widgetmakers.com
B johns@enderson.com
B janes@enderson.com

```

- *Note* that the message text (including header fields) follow immediately after the recipient addresses.

15 Scripting

This chapter introduces Mailtraq's scripting language and discusses how it can be used to provide extensive customisation and automation. Many examples of scripts are also provided.

The following topics are covered :—

- An Introduction to Scripting
- The Scripting Language
- Using Scripts

Mailtraq is a relatively large system, and it is intended to be used in a wide variety of ways (hopefully including many the developers have never considered). To make Mailtraq as flexible as possible, and to be able to adapt it to any situation, a scripting language has been built into it. The

scripting language allows you to enhance Mailtraq and integrate it with other applications and systems that you use.

Chapter 17 provides a complete description of every function supplied with Mailtraq.

What is Scripting?

It may be easier to describe scripting by what it is not. It is not a macro language. Macro languages allow you to make repeated operations more efficient by creating a macro that represents what you would normally do (at the machine's console), and then having the application repeat it when appropriate. It is also not an application programming language. You cannot build independent applications with scripts.

What you can do with the scripts is use the functions that Mailtraq provides to perform various tasks, access information stored in Mailtraq, and integrate Mailtraq with other applications. You can use scripts in message templates to customise them, and you can use scripts in web-page templates to build web-based applications that execute on the Mailtraq web server.

The scripting language is not as comprehensive as a “real” programming language, but it does at least allow you to link it to code written in such a language when you need functionality not provided within the Mailtraq environment.

What do you need to know?

Most people do not have any interest in learning how to program, and you are never forced to use the scripting in Mailtraq. It is simply an option that you may take if necessary. However, if you do find it necessary to use the Mailtraq scripting, but have never programmed before, then you will find the scripting language relatively simple.

If you have used other programming languages, you may find the scripting a little unusual. However, it does follow most of the common rules of programming, and it does provide many of the familiar programming constructs.

The Scripting Language

Functions

At the heart of the scripting language is the function. A function is a repeatable routine that can generate a result, and can be given parameters to compute. What parameters are required, and what value it returns, will depend on the function itself. Functions can, themselves, be parameters to other functions. Unlike many languages, when passing a function as a parameter you are not passing the result of the function, but the function itself. These 'parameter-functions' are executed only when a result is needed, and can be repeatedly executed if necessary.

In Mailtraq, functions are identified by textual names, followed immediately by parameters enclosed within parentheses. There should be no space between the name of the function and the opening parenthesis,

and even if no parameters are being passed, a pair of parentheses must follow a function name.

Operators

Another important part of the scripting language is the ‘operator’.

Operators are actually quite similar to functions, in that they are given parameters and return a result. However, in order to make them easier to read, operators are placed between their parameters instead of prefixing them.

The statement

$$1 + 2 * 3 / (4 - 5)$$

shows a number of operators in action.

Variables

The scripting language also includes variables. These are portions of data into which the results of functions and operators can be placed. Mailtraq often also inserts data into predefined variables before scripts are called.

Variables do not have to be declared before they are used, they are simply created when first referenced. For example, the statement

$$x := 1 + 2$$

results in the variable *x* holding the value 3. Variables can be used in place of constants anywhere in the scripting language. For example, the statement

$$y := x * 2$$

(when following the previous statement) results in the value 6 being stored in the variable *y*.

Expressions

An expression is simply a piece of the script which can be resolved into some value. Functions, operators, variables and constants are all expressions. Functions and operators take expressions as parameters, and expressions can be assigned to variables.

For example,

```
myfunc(1 * 2, myfunc2(x * y), x)
```

is a perfectly valid statement, even though the parameters to *myfunc()* vary considerably.

Constructs

Constructs are used in most languages to control how and when expressions are evaluated. They are noticeably absent from the Mailtraq scripting language, but because you can pass functions as parameters to other functions, the tasks normally handled by language constructs can be just as easily handled by special functions.

Examples of constructs would include *if...then...else* (which will execute a statement only if a given expression evaluates to true) and *repeat...until* (which repeatedly executes a statement until a given expression evaluates to false).

In Mailtraq, the *if...then...else* construct is handled by the *If()* function, which takes two or three parameters. The first parameter is the expression which will be evaluated in order to determine how to proceed. If the expression evaluates to true, then the second parameter will be evaluated and returned, otherwise the third parameter will be evaluated and returned (if it is given, otherwise no evaluation takes place).

For example :—

```
value := If(number < 2, number * 3, number * 4)
```

If the variable *number* is less than two, then *value* will be assigned the result of *number* * 3, otherwise it will be assigned the result of *number* * 4. The *number* * 3 expression could just as easily include a function, such as *MsgGetLineCount()* * 3.

The Top Level

The script itself is simply a series of expressions, and they could be considered the parameters to a function which returns the concatenation of all of them. The only difference with the top level of the script is that the expressions are separated by semicolons instead of commas.

For example, the script...

```
Header( "subject" );
"\n";
Header( "date" )
```

...would evaluate to a message subject and date (taken from the message header fields) separated by a line break. In normal scripts, this information would simply be discarded (as the result from a script is never passed to anything else). However, in templates (where scripts are embedded amongst the plain text), this information would appear in the place of the script. You should refer to Chapter 16 which discusses this further.

- *Note* that there was no semicolon trailing the last expression. This is because of the underlying nature of the script — the expressions are parameters to another hidden function, and the semicolons simply

separate these parameters. You could equally use commas, but semicolons will make the structure clearer.

Loop Constructs

Another construct-like function is *While()*. This function takes two parameters, an expression that controls the loop, and the expression to be executed repeatedly. This is semantically equivalent to the more traditional *while...do...* loop used in many languages.

Here is an example :—

```
i := 0;
while(i < 10,
  do(
    msgGetLine(i) ++ "\n",
    i := i + 1
  )
)
```

We have also introduced another function: *Do()*. This function takes any number of parameters, and simply concatenates the results of each. This has the effect of allowing several expressions to be placed in a single parameter. In fact, the top level of the script is actually a hidden *Do()* function.

In the above example, the *Do()* function (which returns a specific line from a message with a line break appended) is repeated executed until the value of *i* is no longer less than 10.

- *Note* also how the variable *i* was first used. There was no declaration, a value was simply assigned to it. If the *i := 0;* line was excluded, the value would be undefined and the *While()* function would be unable

to compare the value of *i* to 10. You may be wondering what would have happened, so we'll have a look at error handling.

Error Handling

There are generally two types of errors in most programming languages: syntactic and semantic. Syntactic errors are problems with the syntax of the script. Essentially, the syntax is what the script looks like (as opposed to what it does). For example, if you leave out a comma between two parameters, it is a syntactic error. Mailtraq can spot syntactic errors easily because it has a set of rules to follow, and as long as a script passes these rules (i.e. looks correct) then even if it doesn't work, at least Mailtraq can execute it.

Mailtraq cannot execute a script that has syntactic errors, and to help you identify them, it provides a graphing tool. In the *Scripts and Templates* dialogue (available from the *Options* menu in the console), you will find a 'Graph' button for both templates and scripts. This facility will take a script and draw it as Mailtraq has interpreted the script. Places where Mailtraq has become confused because of syntactic errors are highlighted with the word `<ERROR>`.

Semantic errors, on the other hand, cannot be spotted by Mailtraq. Such errors would include, for example, requesting a message line that does not exist. The script looks as though it is OK, but when it is actually used, no line would be found (and is therefore an error). Mailtraq does what it can to keep operating, so invariably semantic errors are simply ignored. If you refer to a message line that doesn't exist, a *null* value is returned. *Nulls* are equivalent to the number zero, or an empty string (depending on the comparison being used).

Even the best programmers produce semantic errors, so the key is to test all your scripts thoroughly before using them.

Data Types

Mailtraq has no real concept of data types. All values are stored in a similar way, and their interpretation depends on the context in which they are used. For example, if you compare the value “1.0” to “1”, Mailtraq will examine both values, determine that they are both numbers, and compare them numerically (thus indicating that they are equal). If you actually intended to compare the string representations (in which case they are clearly different), then you need to explicitly use the `StrCmp()` function (see page 287). Mailtraq will first try to compare values in their numerical form, and only if one of the values is clearly not a valid number, will it perform a string comparison.

Other than numbers and strings (which are just a series of characters) expressions can also evaluate to *true* and *false* (called *Boolean* values). For example, the *greater-than* operator ($a > b$) will return *true* if the first value is greater than the second, and *false* otherwise. In Mailtraq, these values are represented as the strings “TRUE” and “FALSE”.

Lists

Lists are not really part of the scripting language, but they are used so often that it is worth understanding them. A list is simply a series of strings separated by commas. If a string in a list includes a comma, then the string is surrounded by quotes. For example, the value *Monday, Tuesday, “Wed, Fri”, Saturday* is a list containing four items (the third of which is “Wed, Fri”). The functions `ListItem()` and `ListCount()` are used to access lists. Lists are returned by many function, and the function `For()` has been created specifically to iterate through them.

For example :—

```
For("dayname", "Mon, Tue, Wed, Thu, Fri, Sat, Sun",
    MsgAddLine( "on " ++ dayname ++ " do something")
)
```

This function iterates through each day of the week, inserting that value into the *dayname* variable.

String Handling

Strings are used throughout the Mailtraq scripts, and many functions are supplied to manipulate them. There are also some aspects of strings that are part of the scripting language. Quite often you will need to concatenate (join) two strings together. The ++ operator is used for this purpose. For example, "one " ++ "two" becomes "one two".

You may notice that the quotes are used to identify strings (specifically, strings are enclosed within a matching pair of quotation marks). How do you use quotes inside strings? Mailtraq provides a number of escape characters for such a purpose. A quote can be represented with the \q escape. For example, the string "one \qtwo\q three" becomes one "two" three when added to a message, or when written to a file.

Another escape is \n, which represents a line break. When a string containing line breaks is added to a message, the result is that several lines are added.

Finally, the \l and \r escapes represent the left and right braces ({ and }) respectively. This is useful for templates, where the braces enclose embedded scripts.

Working with Scripts

What can you do with a script? Most often, you will want to modify a message in some way. After all, Mailtraq is essentially a mail server. Scripts can access and modify the header and body of a message, can also create new messages. For this reason, many of the functions available to scripts relate to message manipulation.

Signatures Examples

For example, you may wish to add a signature to all messages sent by certain people. You could use the scripting database functions (described on page 292) to look up the sender's name in a database, and if it exists, add a string to a message.

Here is an example :—

```
If(IsAddressLocal(GetMsgSender()),
  Do(
    name := UserOf(GetMsgSender()),
    If(DBRecordExists("signatures", name),
      MsgAddLine(DBRead("signatures", name, "sig")
    )
  )
)
```

In the Mailtraq databases directory (*database\db*) you would create a file called "signatures" containing something like :—

```
[johns]
sig=\n-- \nJohn Smith, Projects Manager\nExt 133

[jamesw]
sig=\n-- \nJames Woods, Accounts Department\nExt 145
```

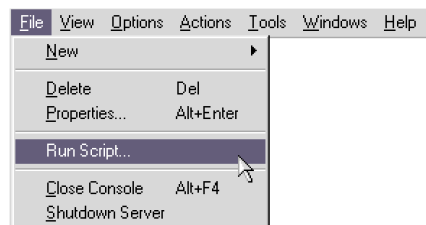

In the next section we will examine how to install and use this script.

Using Scripts

Scripts can be called manually by the user, automatically (either scheduled or as a result of some trigger), or they can be embedded in templates. The templates are discussed in the next chapter, and the following sections will examine the manual and automatic execution of scripts.

Manually Executing a Script

You can execute any script using the *Run Script* command from the *File* menu of the Mailtraq Console. You can also execute a script while Mailtraq is running by running a script file from the Windows Explorer. Files with the extension *.mtq* are associated with Mailtraq's scripting service, and when you double-click on such a file in Explorer, Mailtraq will be sent a message telling it to execute the script.



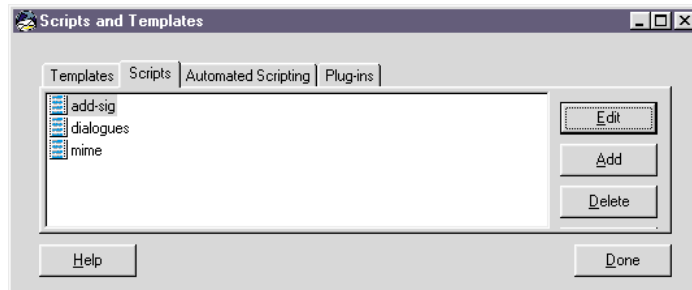
This way of executing scripts is not very useful, as the purpose behind scripts is the ability to repeat them.

Automatically Executing a Script

If you visit the *Scripts and Templates* dialogue (from the *Options* menu in the Mailtraq Console), you will find where the scripts are configured.

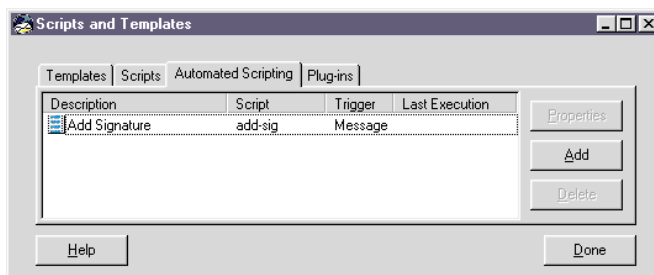
In the Scripts tab, you will see a list of all the scripts that Mailtraq has stored in it's database. To create a new script, click on the Add button. A new file will be added to the list, and you can change the name to something more appropriate. You can then edit the file and enter the contents of the script itself.

The Scripts Tab (Scripts and Templates)



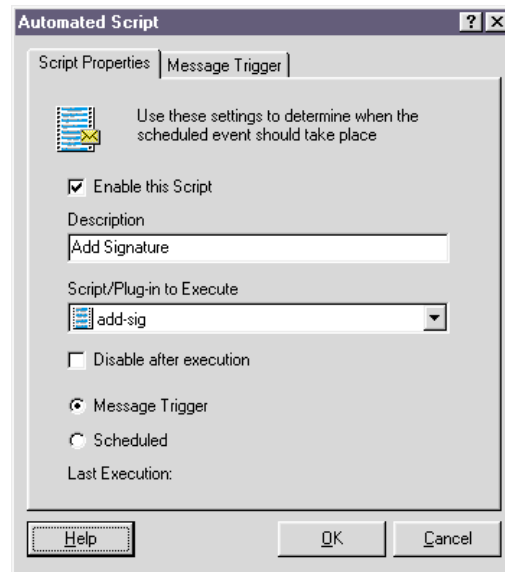
Once you have done this, Mailtraq then 'knows' about the script. If you entered the script described above, and called it 'add-sig', then you could then call that script either from the Automatic Scripting service, or from any other script with the statement *CallScript("add-sig")*.

The Automated Scripts Tab (Scripts and Templates)



The Automatic Scripting tab shows a list of all the automatic events that have been configured. To create a new event, click on the Add button. When a new event appears on the list, you can select it and click on Properties to edit it. In the 'Script/Plug-In to execute' option you can select the previously created script.

Automated Scripts Properties



The Trigger

Automatic Scripts can be triggered either by the receipt of a message that matches a given filter (at a certain point in the Mailtraq routing process), or at a scheduled time. Scripts triggered by the receipt of a message have a number of options.

Firstly, you should specify when the message trigger should take place. This determines at what point during the message handling process Mailtraq examines it's list of scripts and triggers them.

■ Inbound Mail Delivery

This takes place immediately after a message is received (for example, from a gateway or receipt via SMTP). This stage only occurs once per message, and it takes place before the message is routed (i.e., the message recipients will not have yet been altered according to the various routing configurations in Mailtraq).

■ Outbound Mail Delivery

This takes place just prior to the message being delivered. Again, this only happens once per message, and it only takes place if the message actually leaves the Mailtraq server. This means that delivery to local addresses do not trigger scripts with this option.

■ Mail Routing

This can take place several times per message. For example, a message can be routed to it's recipient, which may in turn forward it to other addresses, for which the message will again enter the routing stage. Messages being distributed via a mailing list will always pass through the router at least twice.

■ Mail Storage

This takes place just prior to the message being stored in a user's mailbox (or archive). This is useful in that at this stage the final recipient of the message will be known (as no further routing will take place).

■ Inbound News Delivery

This takes place when a news article is downloaded from another news server, or when a user posts a message from a local news client. At this stage, the news article has not yet been stored in a news group (or placed in the outgoing news folder).

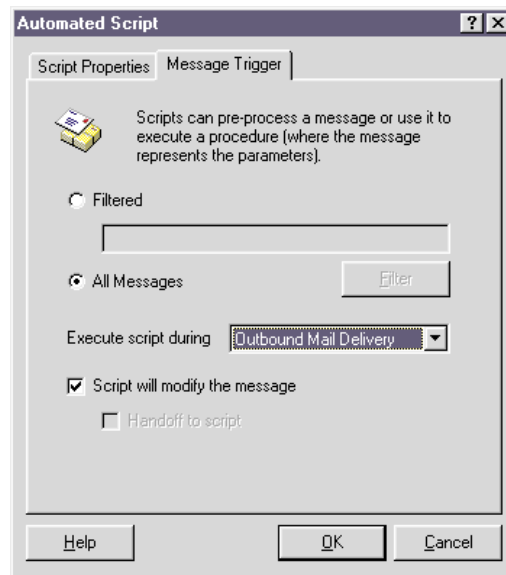
■ Outbound News Delivery

This takes place just prior to a news article being uploaded to a remote news server.

Modifying the Trigger Message

There is also a check-box “Script will modify the message”. Many scripts are intended to alter the message that triggers them in some way (as does the earlier example). Mailtraq needs to know if your script might do that, otherwise it will not wait for the script to complete before continuing with the message delivery.

The Message Trigger Properties



When this options is not selected, the “Handoff to script” option becomes available. If the handoff option is selected, then Mailtraq will not continue delivery of the message. Use this option when you wish to use the message to trigger an event, but when the message itself is of no value after the event has been triggered. For example, a message requesting

data from a database can be automatically answered, and the trigger message discarded.

- *Note* that you should consider keeping the message anyway as a record of the requests. You can always have the mailbox or archive that it is sent to expire the message after a period.

16 Templates

This chapter introduces the concept of Mailtraq Templates, and how they are used to automatically generate messages. The following topics are covered :—

- Introduction to Templates
- Trigger Messages
- Web Page Templates
- An example Web Database using Templates

Introduction to Templates

From time to time Mailtraq will send messages from itself to people. For example, in order to notify a user that a message they sent via Mailtraq could not be delivered. Another example would be a *welcome* message sent out to new subscribers to mailing lists. Imagine if the welcome message could not be customised by the administrators. Mailtraq would only know the name of the list and it would not be able to provide information such as the purpose of the list, and the rules (if any) for posting to it.

On the other hand, take the delivery report mentioned above. If the message was simply custom text written by the administrator, then information specific such as the failed message, to whom it was sent, and the nature of the problem could not be included.

Mailtraq provides a solution to these in the form of *Templates*. These are messages which combine Mailtraq's scripting language with plain text. The format of the template is very straightforward: it is plain text, with the scripting parts enclosed within braces (the "{" and "}" characters).

Here is an example of a template :—

```
Thank you for submitting your message "{ Header("Subject") }".
This message has been distributed to the mailing list.
```

The template is plain text, except for the { *Header("Subject")* } part. When the template is actually used, this will be replaced with the result of the script. In this case, the function *Header()* returns a header field from the current message. Another function will be needed to select the current message, but this will be discussed shortly. For more

comprehensive details on any of the functions in Mailtraq, refer to Chapter 17.

So if this template is triggered by a message with the subject line “Help Please”, then the body of the message generated by the template will contain :—

Thank you for submitting your message “Help Please”.
This message has been distributed to the mailing list.

Using Templates

Most templates can be used under a variety of conditions. For example, a Welcome template might be used by several different mailing lists. How do you tell which list it is being used for? Mailtraq communicates this information to the template by setting various variables prior to actually executing the scripts in a template.

For example, one of the templates supplied with Mailtraq is called *list_welcome*, and this is the *Mailing List Welcome* message. In order to determine which mailing list it applies to, Mailtraq sets the variable “List” to the name of the mailing list.

The example below shows how this could be used to provide the mailing list description :—

```
welcome to {GetSystemVar("\mailslots\" ++ List ++ "\description")}
```

Here the *GetSystemVar()* function is used to obtain information regarding the current configuration of Mailtraq. If the name of the list was ‘sales’,

then the variable requested would be ‘\mailslots\sales\description’, which may contain ‘Sales Team’.

Trigger Messages

Some templates will be used in response to certain messages. For example, the *delivery_failure* template is used to create a delivery report when a message cannot be successfully delivered. When the template is executed, it will have access to two messages: the outgoing message (which is created by the template) and the incoming message which actually triggered the message. In a template, you can use the functions *SelectInMessage()* and *SelectOutMessage()* to switch between them. This is necessary because the message manipulation functions (such as *Header()*, *SetHeader()*, *MsgAddLine()*, etc.) *all* refer to the ‘current’ message.

More Messages

You can also create new messages, and load existing messages from the database. However, these messages only become the ‘current’ message while the *PostMessage()* or *LoadMessage()* functions are in use. This is better described with an example of how the functions are used. Here is an example of the *PostMessage()* function in use :—

```
SelectInMessage();
subject := Header("subject");
sender := AddressOf(Header("from"));
PostMessage("postmaster", "johns",
    Do(
        SetHeader("Subject", "Message from " ++ sender),
        sender ++ " sent the message " ++ subject ++ "\n"
    )
);
SelectOutMessage();
SetHeader("Subject", "Your message: " ++ subject);
```

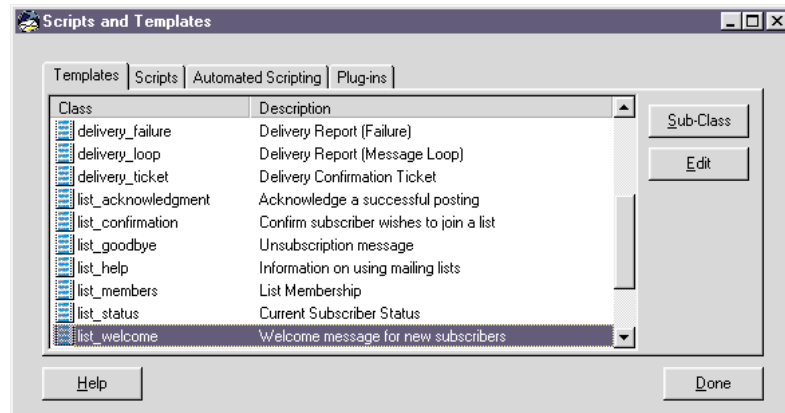
In this example, the subject and sender’s e-mail address is taken from the trigger message and stored in the variables “subject” and “sender” respectively. A new message is then created with the *PostMessage()* command. The *Do()* function within that then executes in the context of the new message. Were the *Header()* functions used within that, they would have referred to the new message, not the trigger. This is why the data was taken from the trigger message first.

After the *PostMessage()* function, the current message is once again the trigger. In order to change the subject of the outgoing message (the one generated by the template), the *SelectOutMessage()* must be used.

Configuring Templates

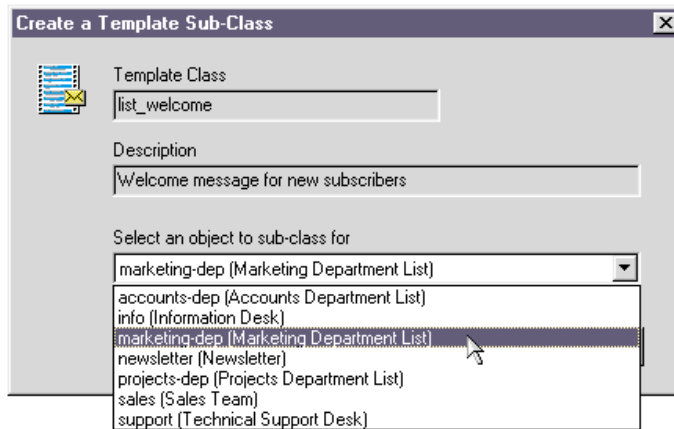
You can configure the templates actually used by Mailtraq in the *Scripts and Templates* dialogue (from the *Options* menu in the Mailtraq Console).

The Mailtraq Templates



In the *Templates* tab, you will see a list of all the template classes available. Many of these templates can be sub-classed for specific objects. For example, the *list_welcome* message (mentioned above) is used by default for all the mailing lists. In most cases you will wish to customise it for each list which will use it, and this is called sub-classing. When you select the *list_welcome* template, the *Sub-Class* button will become enabled.

If you click on this button, you will see a dialogue from which you can select a mailing list. A new template will then appear in the list specifically for that mailing list. Initially, the new template will be a copy of the original template class.

Sub-classing a Template

Web Page Templates

You can also use templates in the Mailtraq Web Server to generate web pages. For details on the Web Server, see Chapter 12. If you refer to page on the Web Server that ends in the extension `.tpl` or `.mtq`, then Mailtraq will assume that it is a template instead of a normal file, and execute it accordingly.

- *Note:* By default, Mailtraq sets the MIME Content-Type of the page to `text/plain`. If you wish to generate an HTML web page, you must set the Content-Type to `text/html`.

Mailtraq web page templates are similar to CGI programs (CGI standing for Common Gateway Interface) used on other web servers. Mailtraq will even take the parameters passed using CGI and place them into variables before the template is executed.

When a web browser requests a web page from the server, it sends a message in a similar structure to an e-mail message. It is therefore possible to access the requesting message using *SelectInMessage()* and functions such as *Header()*.

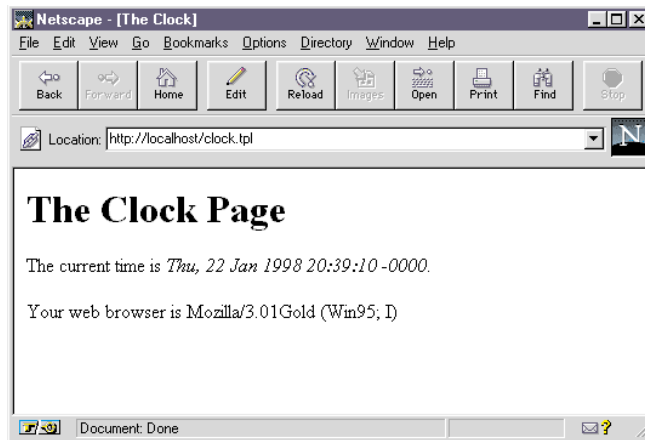
Example Web Template

Here is a simple web page template which simply displays the current time :—

```
{ SelectOutMessage(); SetHeader("Content-Type", "text/html" ) }
<HTML>
  <HEAD>
    <TITLE>The Clock</TITLE>
  </HEAD>
  <BODY>
    <H1>The Clock Page</H1><P>
    The current time is <EM>{ Now() }</EM>.<P>
    Your web browser is {
      SelectInMessage();
      Header("User-Agent") }
  </BODY>
</HTML>
```

The illustration below shows what happens when the page is viewed through a web browser.

*A Web Template shown
in a Web Browser*



A Web Database

The example given above is very simple compared to what can be achieved with Mailtraq. This section will describe a complete web based database using only Mailtraq and the scripting service. In this example, the database contains information about a number of wines, and we will create a web page to access this information.

Firstly, you will need a database table to store the information (which would be stored in the Mailtraq database as *database\db\products*).

Below is an extract from such a file, which shows you what the structure would look like :—

Editing
'database\db\wines'



```

[bot11]
Cultivar=Cabernet Sauvignon
Region=Bulgaria
Vintage=1995
Colour=Red
Price=$6.25
Stock=15

[bot12]
Cultivar=Shiraz
Region=Australia
Vintage=1996
Colour=Red
Price=$5.50
Stock=4

[bot13]
Cultivar=Pinotage
Region=South Africa
Vintage=none
Colour=Red
Price=$5.25
Stock=6

[bot14]
Cultivar=Sauvignon Blanc
Region=France
Vintage=1997
Colour=White
Price=$5.00
Stock=7

```

Next you will need a template to generate the web page that provides access to the information. The template below is then stored in the file *wines.tpl*, which is placed somewhere that can be reached by the web server.

```

{
    SelectOutMessage();
    SetHeader("content-type", "text/html");
    If(cultivar = "", cultivar := "*");
    If(colour = "", colour := "*");
    If(region = "", region := "*");
}
<HTML>
    <HEAD>

```

```

    <TITLE>wine List</TITLE>
</HEAD>
<BODY>
    <H1>wine List</H1><P>
    <FORM METHOD=GET ACTION=wines.tpl>
        Cultivar: <INPUT TYPE=edit NAME=cultivar VALUE={cultivar}><P>
        Region: <INPUT TYPE=edit NAME=region VALUE={region}><P>
        Colour: <INPUT TYPE=edit NAME=colour VALUE={colour}><P>
        <INPUT TYPE=submit>
    </FORM>
    <TABLE BORDER=1>
        <TR><TD>Cultivar<TD>Region<TD>Colour<TD>Price<TD>Stock</TR>
{
    winelist := DBListRecords("wines");
    For(winelist, "wine",
        Do(
            cvar := DBRead("wines", wine, "Cultivar"),
            reg := DBRead("wines", wine, "Region"),
            col := DBRead("wines", wine, "Colour"),
            If(
                wildcardMatch(cvar, cultivar) &
                wildcardMatch(reg, region) &
                wildcardMatch(col, colour),
                Do(
                    "<TR><TD>" ++ cvar ++
                    "<TD>" ++ reg ++
                    "<TD>" ++ col ++
                    "<TD>" ++ DBRead("wines", wine, "price") ++
                    "<TD>" ++ DBRead("wines", wine, "stock") ++
                    "</TR>"
                )
            )
        )
    )
}
</TABLE>

```

```
</BODY>
</HTML>
```

The illustration below shows how the page is viewed from a web browser. The first image shows the page when first visited, and the second page shows the two criteria altered, and the ‘Submit Query’ button pressed.

*The ‘wines.tpl’ template
viewed from a Web
Browser*

Wine List

Cultivar:

Region:

Colour:

Cultivar	Region	Colour	Price	Stock
Cabernet Sauvignon	Bulgaria	Red	\$6.25	15
Shiraz	Australia	Red	\$5.50	4
Pinotage	South Africa	Red	\$5.25	6
Sauvignon Blanc	France	White	\$5.00	7

Wine List

Cultivar:

Region:

Colour:

Cultivar	Region	Colour	Price	Stock
Cabernet Sauvignon	Bulgaria	Red	\$6.25	15
Shiraz	Australia	Red	\$5.50	4

17

Scripting Function Reference

This chapter provides a complete reference for all the functions available in the Mailtraq Scripting Language. Functions added by plug-ins should be documented in the accompanying literature.

To ease reading, all the functions are identified by a pair of empty parentheses following them. For example, the function While will always be written as While()).

Language Constructs

In the Mailtraq Scripting language, functions can be passed as parameters, and they will be executed only when a result is required from them.

Mailtraq doesn't have any language constructs built in, but all of these can be handled with the use of functions as parameters. For example, most languages have a while construct, which repeats the expressions it contains until the evaluator becomes false. Mailtraq instead provides a *While()* function, which takes an expression which it repeatedly evaluates.

Where a single parameter is available, and you wish to pass several functions, the *Do()* function should be used.

Do() This function is essentially the same as the concatenation operator (+ +), but makes the script more readable. *Do()* takes any number of parameters, and returns a concatenation of all of them.

Syntax *result* := *Do*(*expr*[, *expr*[, *expr*...]])

expr An expression to be evaluated.

result The concatenation of all the evaluated expressions given to the function

If() This function simply returns one of two given expressions based upon the result of the first expression. This is similar to the *if...then...else* construct found in most languages. Three expressions are given to the function. If the first expression evaluates to *true*, then the second expression is evaluated and returned, otherwise if a third expression is given, it is evaluated and returned.

Syntax *result* := If(*boolean_expr*, *true_expr*[, *false_expr*])

boolean_expr An expression which should evaluate to *true* or *false*.

true_expr If the *boolean_expr* evaluates to *true*, then this expression is evaluated

false_expr If the *boolean_expr* evaluates to *false*, then this expression is evaluated. If this expression is not given (and the *boolean_expr* evaluates to *false*) then no value is returned.

result The result of the evaluation of either *true_expr* or *false_expr*

While()

This function performs the familiar task of a *while* loop. The content is repeatedly evaluated until the continuation expression evaluates to *false*. Each time the content is evaluated, the result is appended to the function result. This is similar to the *while do ...* construct found in many languages.

Syntax *result* := While(*continuation*, *content*)

continuation An expression evaluated just before each cycle, which will terminate the loop if the result is false

content An expression which is evaluated each cycle loop, with the result being appended to the function output

result The concatenation of the result from each cycle

Repeat()

This function is very similar to the *While()* function, with the exception that the continuation expression is evaluated at the end of each cycle. The loop is terminated when this expression returns true. This is similar to the familiar *repeat ... until* construct found in many languages.

Syntax *result* := Repeat(*content*, *continuation*)

content An expression which is evaluated each cycle loop, with the result being appended to the function output

continuation An expression evaluated after each cycle, which will terminate the loop if the result is true

For() This function iterates through each item in the given list, setting the given variable to the current item each cycle. It is a looping function similar to *While()* and *Repeat()*, where the content expression is evaluated each cycle.

Syntax *result* := For(*list*, *varname*, *content*)

list A comma separated list of items to be iterated through

varname The name of the variable into which the current list item will be placed during each cycle. Note that you should specify the variable's name, not the variable itself. This is done by providing a string such as "name" instead of *name*.

content The expression to be evaluated during each loop cycle

result A concatenation of each cycle's evaluation

Examples The examples below shows each of the constructs in use. Note how the *Do()* function is used to group statements together.

```

num := 10;
while(
    num > 0,
    Do(
        num := num - 1,
        MsgAddLine("Line descending " ++ num)
    )
);
Repeat(
    Do(
        num := num + 1,
        MsgAddLine("Line ascending " ++ num)
    ),
    num < 10
);
For("Mon, Tue, Wed, Thu, Fri", "day",
    If(day != "Fri",
        MsgAddLine("Day " ++ day),
        MsgAddLine("End of the week: " ++ day)
    )
);

```

Another example of the *For()* function is demonstrated below. For each recipient of the trigger message, a reply is posted to the sender.

- Note that the original sender must be stored in a variable before the loop, as the expressions within the *PostMessage()* function are executed in the context of the new message.

```
sender := GetMsgSender();
For(GetMsgRcpts(), "rcpt",
    PostMessage(rcpt, sender,
        Do(
            SetHeader("Subject", "Thank's for your message"),
            "Your message was received by " ++ rcpt
        )
    )
)
```

Define() This function creates a new user-defined function, available to the rest of the script. The function must be defined before it is used, and function definitions can be shared with references to external scripts.

Any number of parameters can be passed to a defined function. From within the function definition, the parameters are referred to by the placeholders \$1, \$2, \$3 etc. The number of parameters is given by \$0.

Syntax *Define(func_name, content)*

func_name The name of the function to be defined (in the form of a string)

content The function expressions. Use the *Do()* function to bind multiple statements.

Example This example shows the definition of a function which will repeat the given string a specified number of times. If a number is not given, the default of '5' will be used.

```
Define("RepeatLine",
    Do(
        local_i := If($0 < 2, 5, $2),
        while(local_i > 0, Do(
            local_i := local_i - 1,
            $1 ++ "\n")
        )
    );
RepeatLine( "Repeated Line", 3);
RepeatLine( "Another Repeated Line")
```

File Handling Functions

These functions are used to manipulate text files, and for reading and writing to them.

OpenFile() Opens a text file for reading. The function returns a handle which identifies this instance of the file, and is valid until it is released with the *CloseFile()* function. The handle also identifies the current file pointer, so that consecutive file reads retrieve consecutive lines of text. When the end of the file is reached, the *EOF()* function will return true.

The same file can be opened by more than one script (and more than once in the same script).

Syntax *handle* := *OpenFile(path)*

path This identifies the file to be opened by its filing system name.

handle This identifier is used in subsequent references to the file, and should be released using *CloseFile()*. The null value is returned if the file could not be opened.

CreateFile()

Creates a new (and empty) text file for writing. The function returns a handle which identifies the file, and is valid until it is released with the *CloseFile()* function. New files can only be written to, as the file pointer is always at the end of the file.

This file cannot be opened by any other scripts until the handle has been released.

Syntax *handle* := *CreateFile(path)*

path This identifies the file to be created by its filing system name.

handle This identifier is used in subsequent references to the file, and should be released with *CloseFile()*. The null value is returned if the file could not be opened.

AppendFile()

This function is similar to *OpenFile()*, except that the file is opened for writing, and the file pointer begins at the end of the current file content.

Syntax *handle* := *AppendFile(path)*

path This identifies the file to be opened by its filing system name.

handle This identifier is used in subsequent references to the file, and should be released with *CloseFile()*. The null value is returned if the file could not be opened.

CloseFile() This function closes a file previously opened with *OpenFile()*, *AppendFile()* or *CreateFile()*. Once a file handle has been closed, the handle no longer has any meaning and should not be used. Mailtraq will automatically close any open files at the end of each script, although it is good practice to explicitly close any files opened for clarity. Further, when a file is open, it cannot be deleted or emptied with *CreateFile()*.

Syntax *CloseFile(handle)*

handle This is the file identifier returned by a call to *OpenFile()*, *AppendFile()* or *CreateFile()*

EOF() This function (End Of File) returns whether or not there is any further data available from the given file. Only files opened with *OpenFile()* can be used with this function. Normally this would be used in a loop in conjunction with the *Read()* function to retrieve all the contents from a file.

Syntax *boolean := EOF(handle)*

boolean *true* if there is more data to be read, or *false* if the end of the file has been reached

handle The file identifier obtained from an *OpenFile()* call

DeleteFile() This function erases the specified file (if it is not currently open). Note that the parameter passed is the name of the file, not a file handle.

Syntax *DeleteFile(path)*

path The filing system name of the file to be deleted

Write() This function writes a string to a file that is currently open for writing. Such files would consist of those opened with *CreateFile()* and *AppendFile()*.

Syntax *Write(handle, string)*

handle A file identifier obtained from a call to *CreateFile()* or *AppendFile()*

string A string of arbitrary length to be written to the file. If the string contains carriage returns, then these will become multiple lines in the file. Note that such lines will require multiple *Read()* statements when retrieving them.

Read() This function reads a single line of text from a text file opened with *OpenFile()*. The file pointer is advanced such that subsequent reads return consecutive lines of text.

Syntax *variable := Read(handle)*

variable A variable name to hold a single line of text retrieved from the file.

handle A file identifier obtained from a call to *OpenFile()*.

Example This example demonstrates a number of file functions. The file “test1.txt” is created, and a line added. This file is then copied to “test2.txt”.

```
h1 := CreateFile("c:\test1.txt");
write(h1, "This is the contents of the text file");
closeFile(h1);
h2 := CreateFile("c:\test2.txt");
h1 := OpenFile("c:\test1.txt");
while(Not(EOF(h1)),
    write(h2, Read(h1))
);
closeFile(h2);
closeFile(h1);
h1 := AppendFile("c:\test2.txt");
write(h1, "This is the second line of the text file");
deleteFile("c:\test1.txt")
```

GetDirectory() This function returns a list of filenames in the given directory. It is also aware of the Mailtraq database, and able to return filenames from the archive volumes which are used for mail slots and POP3 collection accounts. If a fully qualified path is given, then Mailtraq will return the file list obtained from the filing system. If a relative path is given, this is resolved starting with the location of the Mailtraq database.

Syntax *list* := GetDirectory(*path*)

<i>path</i>	The path of the directory
<i>list</i>	A list of filenames found in that directory

Examples These examples show the function being used with a normal filing system directory, a directory in the Mailtraq database, and an archive directory.

```
list := GetDirectory( "c:\windows\temp")

list := GetDirectory( "mail\outbox")

list := GetDirectory( "mail\inbox\johns")

list := GetDirectory( "mail\pop3\enderson@pop3.isp.com")
```

GetDirField() This function obtains the value of a field in the directory record for the given file. The field must be one of those described below, some of which are only applicable to certain file types, and some are only available to files from within the Mailtraq database.

The available fields are

Field	Type	Applicable Directories	Description
size	Integer	Any Directory	The file size in bytes
date	DateTime	Any Directory	The file creation date
subject	String	Any Indexed Directory	The subject line from the message
to	String	Any Indexed Directory	The addressee of the message
from	String	Any Indexed Directory	The sender of the message
serial	String	Any Indexed Directory	The unique serial number of the file
pop3_erased	Boolean	POP3 Collection	Flag indicating if the message was erased
pop3_delete	Boolean	POP3 Collection	Flag indicating if the message is marked for deletion
pop3_fetched	Boolean	POP3 Collection	Flag indicating if the message was downloaded
pop3_download	Boolean	POP3 Collection	Flag indicating if the message is marked for download

- The POP3 Collection Directories contain the message previews, and are in the *mail/pop3/account* (where account is in the *username@host* format).

Syntax *value* := *GetDirField(path, filename, field)*

path Full path of the directory, which may be relative for database directories

filename The filename in the specified directory

field The name of the field to examine

Message Handling Functions

These functions are used to access and modify the current message. The ‘current message’ depends on the type of script being executed, and the context of the function. In scripts triggered by the receipt of messages, then the ‘current message’ will initially refer to the trigger. With templates, the ‘current message’ will initially refer to the template itself (the message being created by the template). Some templates will be triggered by the receipt of messages, and in these cases you can change the current message with the *SelectInMessage()* and *SelectOutMessage()* functions.

SelectInMessage()

This function changes the current message to the trigger message, if one was used. Scripts, Templates and Web Page Templates can be triggered by the receipt of messages.

Syntax *SelectInMessage()*

SelectOutMessage() This function only applies to Templates, which are used to create new messages. Use of this function is actually only useful to Templates and Scripts that are triggered by messages, or Web Page Templates, where the SelectInMessage can be used to change the ‘current message’ to the trigger.

Syntax `SelectOutMessage()`

Header() This function returns the contents of the specified header field in the current message. If the specified header field occurs more than once in the message, the resulting string is the concatenation all the fields.

Syntax `string := Header(field)`

field The field name to extract from the message header

string The text content of the specified field is returned

SetHeader() This function is used to set the specified header field of the current message to the given value. In cases where the specified field occurs more than once, only the first instance of the field is set.

Syntax `SetHeader(field, value)`

field The field name to modify in the message header

value The value to assign to the specified field

DeleteHeader() This function removes a header field from the current message.

Syntax `DeleteHeader(field)`

field The field name to remove from the message header

OriginalMessage()

This function quotes the complete trigger message. A single string is returned which contains both the header fields and the message body. Each line is separated by the ‘\n’ escape, so that when added to another message or written to a file, the line breaks will be maintained.

Syntax `string := OriginalMessage([maxlines])`

maxlines The number of lines to quote (optional). If no parameter is given, the entire message will be returned.

string A single string containing the original (trigger) message.

MsgGetLine()

This function extracts a single line from the body of the current message. The required line is identified by a numerical index, ‘0’ representing the first line. Use the function `MsgGetLineCount()` to determine the number of lines in the message body.

Syntax `string := MsgGetLine(index)`

index The index of the required line, starting from ‘0’

string The requested line from the current message body

MsgInsertLine()

This function inserts a new line into the current message body at the specified location. If the message is five lines in length, then inserting at line ‘5’ will push to current line ‘5’ into position ‘6’.

Syntax *MsgInsertLine(index, string)*

index The line position into which the string will be inserted

string The line to be inserted

MsgAddLine() This function is similar to *MsgInsertLine()*, except that the given string is added to the end of the current message body.

Syntax *MsgAddLine(string)*

string The line to be added to the message

MsgDeleteLine() This function removes the line specified by the given index from the current message body.

Syntax *MsgDeleteLine(index)*

index The index of the line to be removed (starting from '0')

MsgGetLineCount() This function returns the number of lines in the current message body. Use this function to identify the range of valid line indices. For example, a value of '5' indicates that the message lines from '0' to '4' are valid.

Syntax *lines := MsgGetLineCount()*

lines The number of lines in the message body

MsgGetSender() This function returns the sender value from the current message envelope. This is not the same as the *Sender:* or *From:* fields of the message header. See page 71 for details on the message envelopes.

Syntax *sender* := MsgGetSender()

sender The sender value taken from the message envelope

MsgGetRcpts()

This function returns a list of the message recipients specified by the current message envelope. The recipients are not necessarily the contents of the To: or CC: fields. See page 71 for further information on the message envelopes.

Syntax *list* := MsgGetRcpts()

list A comma separated list of the message recipients. Local recipients may not have the domain part of the e-mail address specified.

MsgSetRcpts()

This function replaces the recipients in the current message envelope with the given list. Use this function in scripts that modify trigger messages in order to redirect them. To add a recipient to the message, simply use `MsgSetRcpts(MsgGetRcpts() ++ “, ” ++ newrcpt)`. Note that scripts with the ‘handoff’ property set are also responsible for actually sending the current message, as once the script has been executed, Mailtraq will dispose of it.

Syntax `MsgSetRcpts(rcptlist)`

rcptlist A comma separated list of the recipients to which the current message is to be sent

PostMessage()

This function is used to create a new message and send it to the specified recipients. The content parameter represents the expression to execute in the context of the new message. In other words, when the content is

executed, the ‘current message’ points to the newly created message, prior to it being delivered. The result from the expression is automatically added to the end of the new message body.

If you need more than one statement to be executed, you should bind them with the *Do()* function (as described on page 252).

Syntax

PostMessage(*sender, rcpts, content*)

<i>sender</i>	The value to place in the envelope’s sender field
<i>rcpts</i>	The e-mail addresses to which the message is to be sent
<i>content</i>	The expression to be executed in the context of the new message

Examples

The example below returns a message to the sender of the trigger message indicating that the e-mail address they used was not found.

Note that the *MsgGetSender()* function is executed in the context of the trigger message, while the *SetHeader()* function (which is part of the ‘content’ parameter) is executed in the context of the new message.

```
PostMessage("postmaster", MsgGetSender(),
Do(
    SetHeader("Subject", "Address unavailable" ),
    MsgAddLine("The address you specified was not found."),
    MsgAddLine("Your original message is quoted below :-"),
    MsgAddLine("--"),
    MsgAddLine(OriginalMessage(25))
)
)
```

The example below shows how the value of the ‘content’ parameter is added to the new message. Here the content of the text file is read and forms the body of the new message. Note that *While()* returns a

concatenation of its contents, and since the *Read()* function does not return the line break, it must be added explicitly.

```
handle := OpenFile( "c:\message.txt");
PostMessage( "postmaster", "johns",
    while(Not(EOF(handle)),
        Read(handle) ++ "\n"
    )
);
CloseFile(handle)
```

LoadMessage() LoadHeader()

These function loads a message from disk into the 'current message', and you can use the various message functions described above to access the message headers and body content. Messages can be loaded either from the Mailtraq Database, or from a text file stored elsewhere on disk.

Messages from the database must be given a path relative to the database root. For example, to load a message from the mailbox *johns*, use the path 'mail\inbox\johns\ENDR01234567'. You can examine the Mailtraq database to find the paths to use. Files with the .AFV extension are compound files, and you can reference them as though they are directories (as the example above shows).

You can use the *GetDirectory()* function to obtain a list of files in any directory.

The *LoadHeader()* function is the same as *LoadMessage()*, except that the message body is not loaded. This is useful where the message may be very large, but the content is not important.

To load a message from an ordinary text file, simply use a fully qualified path. For example, 'c:\messages\msg01.txt'. Note that you must use the

Mailtraq message format, where the message envelope is included with the message itself. Refer to page 109 for details.

The last parameter to the function is the expression to execute in the context of the newly loaded message. After this function has been called, the message is discarded.

Syntax `LoadMessage(path, content)`

<i>path</i>	The file name of the message. Use a relative name for messages stored in the database, and a fully qualified name for external messages. Do not attempt to load a message from the Mailtraq Database using a fully qualified name, as this may interfere with the database management.
<i>content</i>	The expression to execute in the context of the loaded message. For example, functions such as <i>Header()</i> will retrieve data from this message.

Example This example demonstrates a script to prepare a message for forwarding as an SMS message to a mobile phone or pager (which can only carry 160 characters, and no subject line).

```
LoadMessage( "mail\inbox\johns\" ++ msgid,
  Do(
    i := 0,
    msgc := "",
    while((i < GetMsgLineCount()) & (Length(msgc) < 160),
      msgc := msgc ++ GetMsgLine(i) ++ " ",
      i := i + 1
    )
  )
);
PostMessage( "postmaster", "johnsms@sms.com", msgc)
```

String Manipulation

These functions provide facilities to manipulate strings, lists and other data. Since the scripting language has no concept of data typing, all the expressions can be treated as strings.

LJ()
CJ()
RJ()

Left Justify (LJ), Centre Justify (CJ) and Right Justify (RJ) are the versatile trio of justification functions. These functions pad the given string out to make it a given length. If the given string is longer than the required length, the functions will trim the string appropriately. *LJ()* adds padding characters to the right, *RJ()* adds to the left, and *CJ()* adds padding characters evenly to both sides.

The padding characters are taken sequentially from the given pad string.

Syntax *string* := LJ(*text*, *width* [, *pad*])

<i>text</i>	The string to which the padding characters are to be added.
<i>width</i>	The required width of the resulting string. If this is longer than the length of <i>text</i> then <i>text</i> is reduced to this length, otherwise characters are taken successively from the pad string until the resulting string reaches the required length.
<i>pad</i>	The characters to pad out the string until it reaches the required length
<i>string</i>	The result of the function, which is always <i>width</i> characters long. If this parameter is not given, the default " " is used (a space).

Example The examples below show typical uses of the functions

```
LJ("text", 10, ".");
// "text....."
CJ("text", 10, "-=");
// "--text=="
RJ(".", 10, "abcde");
// "abcdeabcd."
```

Length() This function returns the length (in characters) of the given string.

Syntax *num_chars* := Length(*string*)

string The text to measure

num_chars The number of characters counted in string

WildcardMatch() This function compares a string with a wildcard expression, and returns a boolean value indicating whether the match was successful. The wildcard expression consists of plain text, with the placeholders '*' and '?' to represent a series of unknown characters and a single unknown character respectively. Note the '*' placeholder represents one or more matching characters.

Syntax *match* := WildcardMatch(*string*, *expr*)

string The string to match

expr The wildcard expression to use when matching string

match A boolean value (*true* or *false*) indicating whether the given string matched the wildcard expression

Example This example shows a variety of wildcard expressions in use.

```
true_match := wildcardMatch(
    "The quick brown fox",
    "*quick*"
);
false_match := wildcardMatch(
    "The quick brown fox",
    "*q????k*"
);
true_match := wildcardMatch(
    "The quick brown fox",
    "*fo?"
);
```

SubStr() This function returns a substring taken from the given string. The position and length of the required substring are given as parameters.

Syntax *sub* := SubStr(*string*, *index*, *length*)

<i>string</i>	The string from which a substring is to be extracted
<i>index</i>	The offset of the first character to extract from the string. The index starts at '0' for the first character.
<i>length</i>	The number of characters to extract

Params() This function divides the given string into parts according to the given separator, and returns the indexed part.

Syntax *part* := Params(*string*, *separator*, *index*)

string The text string to extract a partial from
separator The string that separates the parts of the original string
index The offset of the partial to return

Example The examples below show extraction using a single and multiple character separators. The index begins at zero.

Params("Mon/Tue/wed/Thu/Fri", "/", 2) = "wed"

Params("A + B + C + D + E", " + ", 2) = "C"

ListCount() This function is used to determine the number of items in a list. Lists are used frequently in the scripting language, but are nothing more than a string containing items (optionally surrounded by quotes) separated by commas.

Syntax *count* := ListCount(*list*)

list A list of items separated by commas
count The number of items in the list

ListItem() This function extracts a single item from a list, excluding any quotes if they exist. The item numbering starts at '0'.

Syntax *item* := ListItem(*list*, *index*)

list The list of items
index The offset of the required item in the list
item The returned item from the list

StrCmp()

This function compares two strings and returns an integer value indicating which is the greater (alphabetically). If the first string is less than the second, the result is negative. If the second string is greater, the result is positive (greater than zero). The result is only zero if the two strings are the same.

This function should be used in preference to the comparison operators when it is possible that Mailtraq may interpret the parameters as numeric (and thus apply a numeric comparison). For example, “1.0” is the same as “1” when compared numerically, but different when compared textually (using *StrCmp()*).

Syntax

comparison := StrCmp(*str1*, *str2*)

str1 First string to compare

str2 Second string to compare

comparison Integer representation of the difference. The value is less than zero if the first string is less than the second. The value is greater than zero if the second string is greater. The value is zero when the two strings are the same

DateTimeToReal()**DateToReal()****TimeToReal()**

These functions convert dates and times from Mailtraq’s system formats (*dd/mm/yyyy* and *hh:mm:ss*) to real (floating point) numbers. The integer part represent the number of days since 1900, while the fraction represents the time of the day.

This format is very useful as you can easily compare and perform mathematical operations easily with dates. For example, to find out the number of days between two dates, you can simply use subtraction. To find the number of hours between two dates, simply divide the difference by (1/24).

Syntax *real* := DateTimeToReal(*datetime*)

datetime A date in string format

real The floating point equivalent of the given date

Example This example shows how the numeric form of dates can be used.

```
num := DateTimeToReal("20/10/1997 13:12:11");
if(num > 35723.5501385995,
    days := num - DateToReal("10/10/1997")
)
```

**RealToDateTime()
RealToDate()
RealToTime()**

These functions simply convert real numbers into their equivalent date/time text formats. The FormatDateTime() function provides more versatility.

Syntax *datetime* := RealToDateTime(*real*)

real A floating point value representing some date

datetime The textual version of the date (in one of the Mailtraq system formats)

FormatDateTime

This function takes a date in its numerical format (see *DateTimeToReal()*) and converts it to a user-defined text format. Note that unlike *RealToDateTime()*, Mailtraq will use the locale settings on the host machine. This will result in day names, for example, being generated in the language that is specified in the system locale.

Syntax *text* := FormatDateTime(*format*, *datetime*)

<i>format</i>	A string specifying the format of the date (using the specifiers shown below)
<i>datetime</i>	A number representing a date/time
<i>text</i>	The formatted version of the date

The table below shows the available format specifiers. Other characters will be taken literally (see example).

Specifier	Description
d	A single digit day number (1-31)
dd	Double digit day number (01-31)
ddd	A three letter abbreviation of the day (e.g. Mon, Tue)
dddd	A full length day name (e.g. Monday, Tuesday)
m	A single digit month number (1-12)
mm	Double digit month number (01-12)
mmm	A three letter abbreviation of the month (e.g. Jan, Feb)
mmmm	A full length month name (e.g. January, February)
yy	A two digit year (e.g. 98)
yyyy	A four digit year (e.g. 1998)
h	The hour without a leading zero
hh	Two digit hour (with leading zero)
n	The minute without a leading zero
nn	Two digit minute (with leading zero)
s	The second without a leading zero
ss	Two digit second (with leading zero)
am/pm	Shows either "am" or "pm" (using the string from the current locale). The preceding time also becomes 12 hour.
a/p	Shows either "a" or "p". The preceding time becomes 12 hour.

/	Uses the locale's date separator
:	Uses the locale's time separator
'x'	Text enclosed within single quotes is included literally

Example These examples show a variety of date formats.

```
FormatDate("hh:nn:ss", date)
```

```
FormatDate("dddd, mmmm d yyyy 'at' hh:nn:ss", date)
```

```
FormatDate("ham/pm 'precisely' ", date)
```

AddressOf()

This function takes a string and attempts to extract an e-mail address from it. It is usually used in conjunction with message header fields, where other information (such as the user's name) is also included.

Syntax *email* := AddressOf(*string*)

string Any text string, such as the contents of the From: field

email The first e-mail address found in the string

UserOf()

This function takes an e-mail address and extracts the user part (the string that appears before the '@' symbol).

Syntax *username* := UserOf(*string*)

string Any text string containing an e-mail address

username The user part of the e-mail address

HostOf()

This function takes an e-mail address and extracts the host part (the string that appears after the '@' symbol).

Syntax *host* := HostOf(*string*)

string Any text string containing an e-mail address

host The host part of the e-mail address

NameOf()

This function attempts to extract a person's name from the given string. This is intended to be used in conjunction with the *From:* field, where there are a number of common formats which Mailtraq can interpret (such as "John Smith <johns@enderson.com>", "johns@enderson.com (John Smith)" and so on). In both cases, the result would be "John Smith".

Syntax *name* := NameOf(*string*)

string A text string taken from the *From:* field of a message header

name The extracted name, which will be the e-mail address if a name could not be found

SameAddress()

This function takes two e-mail addresses and determines if they are the same, taking into account the domain aliases.

Syntax *boolean* := SameAddress(*email1*, *email2*)

email1 An e-mail address

email2 Another e-mail address

boolean The result of the comparison

Database Functions

In order to assist the development of powerful scripts, Mailtraq provides a number of database functions. These functions, are designed to provide shared access to data between any number of scripts operating simultaneously. The databases tables are user editable, and are stored in the Mailtraq database under the 'db' directory. The file structure used is the familiar Windows .INI format.

The structure of a
database table

Database Table →

Messages	Sent	Received	Forwarded	Returned	Autoreplied	Copied	Stored
johns							
janes							
mattk							
philip							
samj							
juliej							
piercem							
patricke							
jamesw							
maryt							

Records →

Fields →

The illustration above shows the structure of a database table. The functions in this section are used to access data stored in this type of structure.

Each database table is identified by a unique name, and it is created when it is first referenced. Each table contains a number of records, again created when first referenced. Each record is referred to by a unique name, set by the script. Each record can have any number of fields, and each field stores a single value. Data is retrieved with the *DBRead()* function, and written with the *DBWrite()* function.

DBRead() This function returns the current contents of the specified field, record and database table. An empty string is returned if the field does not exist, and the database table will be created if it has not yet been created either.

Syntax *value* := DBRead(*database*, *record*, *field*)

<i>database</i>	The name of the database table. Note that this is essentially the filename, so operating system file naming conventions should be considered. For example, the database name is not case-sensitive, and certain characters are not permitted.
<i>record</i>	This is the name of the record, in the form of a string. The name is not case sensitive.
<i>field</i>	This is the name of the field from which a value is to be retrieved. The field name is not case sensitive. Records do not have to share the same set of fields, so it is in no way inefficient to use different field names in different records.
<i>value</i>	The string of data extracted from the database. If the field, record or table do not exist, an empty string will be returned.

DBWrite() This function writes the given value into the database. If the specified database table, record or field does not exist, they will be created.

Syntax `DBWrite(database, record, field, value)`

<i>database</i>	The name of the database table. Note that this is essentially the filename, so operating system file naming conventions should be considered. For example, the database name is not case-sensitive, and certain characters are not permitted.
<i>record</i>	This is the name of the record, in the form of a string. The name is not case sensitive.
<i>field</i>	This is the name of the field from which a value is to be retrieved. The field name is not case sensitive. Records do not have to share the same set of fields, so it is in no way inefficient to use different field names in different records.
<i>value</i>	The string of data to be written into the database. The string should not be longer than 200 characters, and it should not contain line breaks. For these reasons, it is not possible to store messages or parts of messages in the database. It is, however, possible to store the value of header fields in this way.

DBRecordExists() This function checks whether or not the given database table and record exist.

Syntax `exists := DBRecordExists(database, record)`

<i>database</i>	The name of the database to examine
<i>record</i>	The record to search for in the given database
<i>exists</i>	The result is <i>true</i> if the record and database do exist, and <i>false</i> if either is missing

DBDeleteRecord() This function removes the specified record from the specified database.

Syntax `DBDeleteRecord(database, record)`

database The name of the database to delete a record from

record The name of the record to remove. If this record does not exist, no action is taken.

Mailtraq System Access

Mailtraq provides a means of accessing the system configuration by representing all the system settings as a hierarchical tree of keys (similar to the system registry). Each key contains a number of variables that represent configuration settings. The current contents of any variable can be read with the `GetSystemVar()` function, and written to with the `SetSystemVar()` function. Some variables are write-only and act as functions when values are assigned to them.

Objects in Mailtraq (such as Mail Slots) are represented by branches of the tree. For example, to access the description of a mailbox called “johns”, use the variable “mailslots/johns/description”.

The System Tree

The keys are arranged in a hierarchical structure, as shown below.

Key	Type of values found in the key
/	General Variables representing the entire system
/mailslots/slot/	The Mail Slot properties
/mailslots/slot/subscribers/sub/	The Subscriber properties, for mailing lists
/mailgroup/group/	Mail Group properties
/pop3/account/	POP3 Collection account properties

The following tables show all the variables available to the scripting system. The *Type* column indicates the variable type that should be given to *SetSystemVar()* and the type returned with *GetSystemVar()*. These items may appear :—

(W)	The variable is write-only (i.e. a function is performed when the variable is written to)
String	The value is a textual string
Integer	The value can always be interpreted as an integer number
Boolean	The value will either be TRUE or FALSE
DateTime	The value will be a date and time using the Mailtraq system format (dd/mm/yyyy hh:mm:ss)
Time	The value will be a time in the format (hh:mm:ss)
Option	The value must be one of the described strings
List	The value is a comma separated list. Each item in the list may be enclosed within quotation marks.

General System Variables (/)

These variables apply to the entire system, rather than a specific object such as a Mail Slot.

Variable Name	Type	Purpose
AddArchive	String (W)	Create a new archive with the given name
AddMailbox	String (W)	Create a new mailbox with the given name
AddMailingList	String (W)	Create a new mailing list with the given name
BounceOption	Option	How undelivered mail is handled <ul style="list-style-type: none"> "returntosender" Return (refuse to accept) "returnreport" Create a failure report "forwarduser" Send to a specific mailbox "readdress" Re-Address to another domain "create" Create a new mailbox "sendtooutbox" Put in the outbox
BounceSite	String	Site for BounceOption=readdress
BounceUser	String	Address for BounceOption=forwarduser
BreakIntoPartials	Boolean	Enable/Disable the Break Incoming Mail option
ForwardIncoming	Boolean	Forward all incoming mail to a local mailbox
ForwardIncomingTo	String	Forward all incoming mail to this mailbox
HostAliases	List	Specifies the domain aliases (comma separated)
LocalAreaNetwork	List	The IP Addresses representing the local network
LocalHost	String	Specifies the local domain name
MailboxTemplate	String	Mailbox to copy for BounceOption=create
MailSortingRules	List	The rules used for the Mail Sorting facility
MaximumMessageSize	Integer	The maximum size of incoming mail to accept, or the size which mail is broken into
NetworkOption	Option	Specifies the way Mailtraq handles Internet Dialup <ul style="list-style-type: none"> "dialup" Use DUN to connect "direct" Permanent connection "sync" Synchronised Times "virtual" Virtual Network "none" No Internet Connection
OutgoingForward	Boolean	Enable/Disable the option to copy outgoing mail to a local mailbox

OutgoingForwardUser	String	The username to forward copies of outgoing mail to
RASentry	String	Current DUN connection
RAScheduleEnable	Boolean	Enable/Disable dial-up schedules
RemoteSMTP	String	The outbound mail server
RemoveMailslot	String (W)	Delete the specified mail slot
SenderBarring	List	List of sender addresses to bar
SenderBarringMessage	String	The reason to give when an address is barred
SenderBarringRcpt	List	List of recipient addresses to bar

Mail Slots (/mailslots/slot)

These variables provide access to the Mail Slots in Mailtraq. Mail Slots are referred to by their name (identifier). For example, */mailslots/johns/description* may return “John Smith”. Note that many variables only apply to certain types of Mail Slot.

Variable	Type	Purpose of Variable
Password	String	Mailbox POP3 Password (Mailboxes)
StoreMail	Boolean	Enable/Disable the storing of mail in mailboxes (Mailboxes)
EnableForwarding	Boolean	Enable mailbox forwarding (Mailboxes)
ForwardTo	List	Mailbox forwarding address (Mailboxes)
Members	List	Archive membership (Archives)
Administrators	List	Archive administrators (Archives)
Aliases	List	Mail Slot aliases
AddSubscriber	String	Add a new list subscriber by specifying their e-mail address (Lists)
arDefault	Boolean	Enable the default message (Archives)
arIndexDate	Boolean	Include dates in the index (Archives)
arIndexSender	Boolean	Include the sender in the index (Archives)
arIndexSize	Boolean	Include sizes in the index (Archives)
arLimitAdministration	Boolean	Limit Administration (Archives)
arLimitMembership	Boolean	Limit Membership (Archives)

arRequestByProxy	Boolean	Enable requests by proxy (Archives)
AutoresponderText	String	The Autoresponder message (Mailboxes)
DefaultMessage	String	Default message for Archives (Archives)
Description	String	Descriptive name of the mail slot
DigestSize	Integer	Size (in kb) to trigger digest delivery (Lists)
DigestTime	Integer	Number of days between digest deliveries (Lists)
ExpireForward	String	Address to forward expired mail to (Archives and Mailboxes)
ExpireHours	Integer	Number of hours after which to expire messages (Archives and Mailboxes)
Filter	String	Mail Slot filter
GatewayDirectory	String	Directory for mail gateway (Mailboxes)
Maintainer	String	Mailing List maintainer (Lists)
mbAutoresponder	Boolean	Enable the autoresponder (Mailboxes)
mbExpireForward	Boolean	Forward expired mail (Mailboxes)
mbExpireMail	Boolean	Expire mail (Mailboxes)
mbExpiryNotify	Boolean	Notify sender on expiry (Mailboxes)
mbGatewayKA9QQueue	Boolean	Use KA9Q Queue mode for the gateway (Mailboxes)
mbGatewayKA9QPegasus	Boolean	Use Pegasus mode for the gateway (Mailboxes)
mbGatewayKA9QRoute	Boolean	Use KA9Q Route mode for the gateway (Mailboxes)
mlAddTag	Boolean	Add the tag specified in "Tag" variable (Lists)
mlAddTagFile	Boolean	Add the tag file specified in "TagFile" variable (Lists)
mlAnonymous	Boolean	Anonymous Option (Lists)
mlConfirmSubscription	Boolean	Confirm subscription-by-mail (Lists)
mlDigest	Boolean	Enable digests (Lists)
mlDigestContents	Boolean	Include a contents table in a digest (Lists)
mlDigestSize	Boolean	Enable size triggered digests (Lists)
mlDigestTime	Boolean	Enable scheduled digests (Lists)
mlEvenDistribution	Boolean	Distribute mail evenly between subscribers (Lists)
mlModerated	Boolean	List is moderated (Lists)
mlProxylsAdministrator	Boolean	Proxy must be administrator (Lists)
mlReplyToAuthor	Boolean	Set Reply-To to the message author (Lists)

mlReplyToDefault	Boolean	Leave the Reply-To of the message unchanged (Lists)
mlReplyToList	Boolean	Set Reply-To to the List (Lists)
mlReplyToOther	Boolean	Set Reply-To to the "ReplyTo" variable (Lists)
mlSecure	Boolean	Only accept subscription-by-mail commands accompanied by the subscriber's password (Lists)
mlStripDeliveryTicket	Boolean	Remove the delivery ticket from messages (Lists)
mlSubscriptionByMail	Boolean	Subscription By Mail (Lists)
mlSubscriptionByProxy	Boolean	Subscription By Proxy (Lists)
Moderator	String	Mailing List moderator (Lists)
NextDigest	DateTime	Next Digest Delivery (Lists)
RemoveSubscriber	String	Remove a list subscriber by specifying their e-mail address (Lists)
ReplyTo	String	Reply-To address (Lists)
Tag	String	Message Tag text (Lists)
TagFile	String	Filename for message tag (Lists)

Subscriber Properties (/mailslot/slot/ subscribers/sub)

The variables shown in the table below provide access to the subscriber properties in a Mailing List. The subscriber is referenced by their e-mail address (the *sending* address, not the *reply-to* address). The *default* and *guest* subscription properties can be referred to through the reserved addresses ".default" and ".guest" respectively.

Variable	Type	Purpose of Variable
Details	String	Additional details (usually organisation name)
EMail	String	E-Mail address of subscriber (Key)
Filter	String	Message Filter
JoinDate	DateTime	Joining date
Name	String	Full Name of subscriber
Password	String	Password
PostDate	DateTime	Last time subscriber posted to list

ReplyTo	String	Reply-To address for subscriber
sfAdmin	Boolean	Subscriber is classed as an administrator
sfBarred	Boolean	Mail from subscriber is always refused
sfConceal	Boolean	Subscriber wishes to remain anonymous
sfDigest	Boolean	Subscriber receives Digests (multipart/mixed format)
sfDigestMD	Boolean	Subscriber receives Digests (multipart/digest format)
sfDigestText	Boolean	Subscriber receives Digests (text/plain format)
sfModerated	Boolean	Mail from subscriber is accepted, but held for moderation
sfPublic	Boolean	Subscriber wishes to appear in the membership list
sfRecBin	Boolean	Subscriber accepts messages with binary attachments
sfRecipient	Boolean	Subscriber receives mail
sfSender	Boolean	Subscriber is allowed to post
sfSndBin	Boolean	Subscriber is permitted to post binary attachments
sfSuspended	Boolean	Suspended flag
Volume	Integer	Number of messages posted

Mail Groups (/mailgroups/group)

These variables provide access to the Mail Groups, which represent the logical Mail Slot groupings and act as an interface to the accounting facilities. The Mail Groups are accessed with the *group name*.

Variable	Type	Purpose of Variable
AccountingPeriod	Option	Length of accounting period "daily" "weekly" "monthly" "annually"
Description	String	Mailgroup description
EnableAccounting	Boolean	Enable the accounting settings
EnableMailboxLimit	Boolean	Limit the size of the mailbox
EnableSendLimit	Boolean	Enable limits on messages sent by the mail slot address

InheritAccounting	Boolean	Inherit parent's accounting settings
LastPeriod	DateTime	Date this accounting period began
MailboxLimit	Integer	Maximum size of the mailbox
Name	String	Mailgroup name
NextPeriod	DateTime	Date of next accounting period
Parent	Integer	ID of parent mailgroup
ReportTo	String	Address to which the report is sent
SendLimitMessages	Integer	Maximum number of messages per period
SendLimitMessageSize	Integer	Maximum individual message size (kb)
SendLimitVolume	Integer	Maximum volume of mail (kb) per period
SendReport	Boolean	Send a report by mail
SendWarning	Boolean	Send a warning message to the mail slot when limits are being reached
Warning	Integer	Send the warning when the volume/messages reach the given percentage of the limit

POP3 Collection Account Properties (pop3accounts/account)

These variables provide access to the POP3 Collection facilities. The accounts are referred to by a key created by combining the user name and the POP3 server with the '@' symbol (e.g. *enderson@pop3.isp.com*).

Variable	Type	Purpose of Variable
Active	Boolean	Enable/disable account
Account	String	The POP3 account (username@host format)
Password	String	The login password
Local	Boolean	Is the account on the local network?
LastAttempt	DateTime	Last time Mailtraq attempted to connect to the account
LastCollection	DateTime	Last time Mailtraq connected to the account
LeaveOnServer	Boolean	Leave the mail on the server after collection

RecipientOption	Option	Option for handling extracted recipients "ignorenonlocal" "assumelocal" "forwardnonlocal"
EnableFilter	Boolean	Enable/disable the message and size filter
Filter	String	The message filter
FilterSize	Integer	The maximum size of message to collect
Frequency	Integer	Minimum period (in minutes) between collection attempts
Routing	Boolean	Route mail based on an extracted address?
RouteOption	Option	Method to use for routing "fields" Extract from specified fields "env" Use ENV command "received" Extract from Received: field
Recipients	List	Recipients to send to if not routing
SendSummary	Boolean	Send a summary message?
SummaryTo	String	Address to send the summary message to
RcptFields	List	List of fields to examine for recipient addresses (if RouteOption="fields")
AutoDelete	Boolean	Automatically delete messages over a certain age
AutoDeleteAfter	Integer	Number of days after which to delete the messages

GetSystemVar() This function returns the current value of the specified variable. The above system tree describes the available variables.

Syntax *value* := GetSystemVar(*variable*)

variable Full path (e.g. "/mailslots/johns/description")

value The current contents of the variable

SetSystemVar() This function assigned a value to the specified variable. The above system tree describes the available variables. Some variables (such as

/addmailbox) are write-only, and cause an action to take place when the value is written to them.

Syntax **SetSystemVar(*variable*, *value*)**

variable The full variable path (e.g. *"/mailslots/johns/descriptions"*)

value The value to be placed into the given variable

ConsoleMenuCommand()

This function scans the Mailtraq Console menu for a matching entry, and then executes the associated function (as though the user clicked on the menu item).

Syntax **ConsoleMenuCommand(*menu*)**

menu The text taken from a menu entry

Examples

These examples show how the menu entry names only have to match alphanumeric characters.

```
ConsoleMenuCommand("Go Online")
```

```
ConsoleMenuCommand("Outgoing Mail...")
```

```
ConsoleMenuCommand("IncomingMail")
```

External Programs

Mailtraq can be extended with outside programs and function libraries using the scripting language. The following functions allow you to call custom made applications, and integrate Mailtraq with existing software.

By writing external functions, it is possible to develop more complex extensions to Mailtraq than those possible with the scripting language.

Execute() ExecuteAsync()

These functions call an external application with the given parameters. The *Execute()* function will suspend the script in which it is used until the application terminates. Conversely, the *ExecuteAsync()* function allows the script to continue immediately.

Although it is possible to send information to the executed application via the ‘command line parameters’, there is no means for the application to pass information back to the script. However, it may be useful for the application to store information in an external file in a known location, perhaps one passed as a parameter, which the script can then read from. If you choose to use this method, then you should not use the *ExecuteAsync()* function, as you cannot be certain the application has completed writing its output file by the time you read it.

Syntax

Execute(application, [parameters])

application The full path name of the application to execute

parameters The parameters, separated by spaces, passed to the application. These are the same as the command-line parameters that might be given if the application was executed from the Windows Shell.

Example

This example runs the Notepad text editor (assuming that Windows has been installed into the “c:\windows” directory) and gives it the name of the file to open. A more effective text editing example is given with *ShellExecute()*.

```
Execute("c:\windows\notepad.exe", "c:\windows\temp\info.txt")
```

ShellExecute()

This function is very similar to *ExecuteAsync()*, in that it is used to run a program without waiting for it to terminate. However, this command concentrates on files (or documents), rather than the applications used with them. Essentially, the given parameters are passed to the Windows Shell to be handled, rather than by calling the application directly. This also makes it possible for the Shell to select an appropriate program to run for a given filename. For example, given the filename of a text document, the Shell may open it using *Notepad* or *Wordpad*. You can also run specific programs using this function, but you do not need to know their filenames (providing the Shell can locate them).

Syntax

ShellExecute(*filename*, [*parameters*])

filename The filename of the document to be opened, or the application to be executed. If the filename is an application in the Windows Path, you do not need to give a full path-name for it.

parameters Optional parameters given to the application eventually called by the Shell.

Example

This example is similar to that shown in *Execute()*. The advantage is that you don't need to know where the text editor is located, or even what text editor has been registered. Note that the Shell will select an application based upon the file's extension (i.e. ".txt" in this example). If you wish to edit a text file that does not end in this extension, you will need to specify the application (as shown in the second example).

```
ShellExecute("c:\windows\temp\info.txt")
```

```
ShellExecute("notepad", "c:\windows\temp\info.msg")
```

CallDLLFunction()

This is one of the most important functions in the Mailtraq scripting language. Using this function, you can integrate Mailtraq with other

software, or provide complex added functionality to Mailtraq that would not be possible with the scripting functions alone. This function is also a key part of the Mailtraq Plug-In mechanism.

CallDLLFunction() simply calls another user-defined function stored in a Dynamic Link Library (DLL). The user-defined function must be exported from the library in the same way as any .DLL function (i.e. standard 'C' type parameters).

Communication between Mailtraq and the DLL function take place through a shared block of memory (allocated and released by Mailtraq). The DLL function must return a single 32-bit integer (which must contain zero), and take two parameters: a 32-bit integer defining the size of the memory block, followed by a 32-bit pointer to the start of that block.

'C' Prototype

```
int FunctionName(int size, char *ptr);
```

Pascal Prototype

```
function FunctionName(size: Integer; ptr: PChar): Integer; stdcall;
```

- Note the use of the keyword *stdcall* in the Pascal prototype.

Mailtraq passes the parameters given in the *CallDLLFunction()* in the form of a null-terminated string, entered into the shared memory block. When the function is completed, the contents of the memory block is assumed to be a null-terminated string, and is returned as the result of the *CallDLLFunction()*.

It may be worthwhile to bind regularly used functions to scripting functions using *Define()*. These function definitions can be stored in a

single file that can be called by any script that wishes to use the library functions.

Syntax *result* := CallDLLFunction(*library*, *function*, [*parameters*], [*datasize*])

<i>library</i>	The full path name of the DLL that will contain the requested function
<i>function</i>	The name of the function to execute in the DLL. This should be the exported name of the function.
<i>parameters</i>	This optional parameter is a text string that is placed in the shared data-block prior to the DLL function being executed.
<i>datasize</i>	This optional parameter determines how large the memory block will be. This is also the maximum size of the data that can be passed between Mailtraq and the DLL function. If this value is not given, the default of 256 will be used.
<i>result</i>	This is the text string contained in the memory block after the function has been executed. If the function does not modify the memory block, then this will be the same as the <i>parameters</i> value.

Example This example shows how two external DLL functions (for accessing registry data) are bound to more user-friendly Mailtraq functions. The definitions could have been stored in a separate file, called with the *CallScript()* function at the start of any script that might need them.

```
Func_Lib := "c:\lib\myfuncs.dll";
Define("RegReadVal",
    CallDLLFunction(Func_Lib, "RegReadVal", $1, 4096)
);
Define("RegWriteVal",
    CallDLLFunction(Func_Lib, "RegWriteVal", $1 ++ "=" ++ $2, 4096)
);

val := RegReadVal("\Software\MyProgram\Value1");
RegWriteVal("\Software\MyProgram\Value2", val);
```



Glossary

This chapter describes a number of words and terms used in both this manual and in Mailtraq.

Barring

This term refers to a facility where mail can be refused by a mail server based upon the message sender (or recipient). With SMTP, barring is the most efficient form of avoiding unwanted mail as the messages can be refused before they are downloaded.

Bounce

This is a term referring to what happens when a mail server (to which the message is addressed) cannot complete delivery, and returns the message. In Mailtraq, bounced mail does not necessarily have to be returned to the sender.

Client	A client is an application which connects to a server to perform some task. For example, an SMTP client is an application that connects to an SMTP server to deliver an e-mail message.
DNS	Domain Name Server. Every machine on the Internet (and on local networks) has a unique IP address. However, the addresses can be difficult to remember (and therefore easy to enter incorrectly) and the addresses can change. DNS machines translate ordinary textual names into IP addresses.
DUN	Dial-Up Networking. This is a term used (mostly in Windows '95) to refer to a temporary connection made (using a modem) to an Internet Service Provider. Dial-up networking can actually be used to connect any two machines together. <i>See RAS.</i>
Dynamic IP Addresses	Some Internet Service Providers dynamically assign an IP address to customers when they connect. Because these addresses vary, the customer cannot have a permanent DNS record which refers to their machine, and thus connections cannot be established from other parties without first sending them the current IP address.
Finger	This is also called the 'User Information Protocol'. It is a communication standard for retrieving information about users or mailboxes from remote machines. Service Providers often offer finger services to let users determine how much mail is waiting for them, or to kick mail servers into delivering mail. The Finger Server usually resides on TCP/IP port 79.
HTML	HyperText Markup Language. This is an implementation of SGML which has become the standard for defining formatted text in Web pages.

HTTP HyperText Transport Protocol. This is a generic communication protocol for transferring messages between machines. It is primarily used for transferring pages and files on the World Wide Web, although it can in fact transfer any type of data. The HTTP Server usually resides on TCP/IP port 80, although HTTP Servers intended only for proxying requests are often set to listen on TCP/IP port 8080.

IP Address The IP address is used to uniquely identify a machine on a network. No two machines may share the same IP address. An IP address is defined by four numbers (in the range 0 to 255) separated by dots. For example, 192.168.0.1 would be a valid IP address. A single machine may be known by several IP addresses.

MX Resolution MX machines (or Mail Exchanges) are mail servers designated to receive mail for specific domains. MX Resolution involves identifying the appropriate Mail Exchange when delivering mail to a user. MX Records can be retrieved from DNS machines.

NNTP Network News Transfer Protocol. This is a communication protocol for transferring news articles between news servers and clients. Unlike SMTP and POP3, NNTP can both send and receive news articles. The NNTP Server usually resides on TCP/IP port 119.

POP3 Post Office Protocol (Version 3). This is a communication protocol for collecting mail from a remote mail store. The protocol is initiated by a client that wishes to receive mail, and it connects to a POP3 Server to do this. The POP3 Server usually resides on TCP/IP port 110.

Port	This is the TCP/IP term for one end of a network connection. A single machine can have a number of TCP/IP services, each 'listening' on a separate port. For example, SMTP receives connections on port 25, while POP3 receives connections on port 110.
RAS	Remote Access Service. This is a Windows NT facility for managing temporary network connections, such as those handled with a modem. The term is similar to DUN (which is more related to Windows '95).
Server	A machine (or application) offering a service to which clients can connect. A server waits for connections, and is usually able to handle several simultaneous connections.
Smarthosts	A Smarthost is a Mail Exchange that accept mail for any domain, and then it handle the task of identifying the appropriate Mail Exchange for the recipient and passing the message on to it. Most Internet Service Providers supply a smarthost so that mail clients do not need to handle MX Resolution, and can therefore save on online time for mail delivery (particularly when the appropriate Mail Exchanges are temporarily unavailable). Most Service Providers limit access to smarthosts to their customers. Mailtraq's SMTP Server makes the machine running Mailtraq a smarthost (unless configured otherwise).
SMTP	Simple Mail Transfer Protocol. This is a communication protocol for sending e-mail from one machine to another. The protocol is initiated by the machine currently in possession of a message (the SMTP Client) that it wishes to send to another machine (the SMTP Server). The SMTP Server usually resides on TCP/IP port 25.

Static IP Addresses

Static IP Addresses are where connections made to an Internet Service Provider always use the same IP address. Because the IP address is always the same, Domain Name Servers can hold records translating a domain name into that address.

TCP/IP

Transmission Control Protocol / Internet Protocol. TCP/IP is the networking protocol on which the Internet is based. It was designed for wide area networks, but is equally effective on local area networks.



Bibliography

This chapter provides a reference to a number of documents that describe the systems and protocols used by Mailtraq. The information in these documents is the most comprehensive and accurate available.

Many of these documents are RFCs (or Request For Comments). These are documents maintained by the Internet Engineering Task Force (IETF) and form the basis for common Internet standards.

Where possible, these documents have been supplied on the Mailtraq CD, in the *\docs* directory.

RFC821

“Simple Mail Transport Protocol”

August 1982

Jonathan B. Postel

Information Sciences Institute, University of Southern California

This document is the specification for SMTP mail delivery.

RFC822 “Standard for the Format of ARPA Internet Text Messages”

August 1982

David H. Crocker

Department of Electrical Engineering, University of Delaware

This document specifies the structure of e-mail messages. This format is also used for news articles and is the basis for practically every form of Internet message. Amongst other things, it also defines many of the standard header fields.

RFC977 “Network News Transfer Protocol — A Proposed Standard for the Stream-Based Transmission of News”

February 1986

Brian Kantor, Phil Lapsley

U.C. San Diego, U.C. Berkeley

This document is the NNTP standard, which is the protocol Mailtraq uses for all its news services. See [draft-barber-nntp-news-07] for the latest revisions to this standard.

RFC1725 “Post Office Protocol — Version 3”

November 1994

J. Myers, Carnegie Mellon, M. Rose

Dover Beach Consulting, Inc.

This document is the current revision (and standard) for POP3, which is the protocol used by Mailtraq to provide mail clients with access to locally stored mail.

RFC1521 “MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies”
September 1993
N. Borenstein, N. Freed
Bellcore, Innosoft

This document is the standard for MIME messages, which is the basis for carrying non-standard information in Internet messages.

RFC1288 “The Finger User Information Protocol”
December 1991
D. Zimmerman
Center for Discrete Mathematics and Theoretical Computer Science

This is the standard for the Finger protocol.

RFC1522 “MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text”
September 1993
K. Moore
University of Tennessee

This document describes the means by which information types and formats are identified in MIME messages.

RFC2068 “Hypertext Transfer Protocol — HTTP/1.1”

January 1997

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee

U.C. Irvine, Digital Equipment Corporation, MIT/LCS

This is the standard for HTTP (version 1.1) which is the basis for Web File transfers. Mailtraq supports HTTP/1.1, but also makes some client requests in the form of HTTP/1.0.

draft-barber-nntp “Common NNTP Extensions”

S. Barber

Academ Consulting Services

This Internet Draft provides details on many of the current extensions to the NNTP system (RFC977). All the extensions implemented by Mailtraq are documented in this draft.

Index

A

Accounting	92
Active Connections.	45
Address Re-Writing	102
Administration	
<i>Administration Topics</i>	121
<i>Web Administrator</i>	124
Advanced Routing Table.	110
Aliases	85
Announcements	161
Archives	
<i>ADD Command</i>	172
<i>DIGEST Command</i>	172
<i>INDEX Command</i>	171
<i>Introduction</i>	169
<i>SEND Command</i>	171
Article Threads.	147
Automated Scripts.	246
Automatic Schedules	68
Autoresponders	89

B

Bandwidth	142
Barring	101,313
Bouncing.	99,313
Break Into Partials	107 - 108
Browsers	177

C

Carbon Copy Recipients	116
CGI	185
Clearing Host Assignments	111
Clipboard	131
Configure Columns	67
Confirming Subscriptions	164
Console	39
Control Messages	163
Control Panel.	40
Crawlers	
<i>See Web Crawlers</i>	

D

Database	
<i>Location of Database</i>	134
Default News Server	144
Delivering Service Providers	7
Delivery Hops	6
Delivery Reports	100
Detailed Message Descriptions	130
Dial Up Networking	
<i>Direct Connections</i>	55
<i>Gettign On Line</i>	13
<i>Leased Lines</i>	55
<i>Overview</i>	53,57
Dialogues	
<i>Archives</i>	169
<i>Automated Scripts</i>	248

<i>Dial-Up</i>	61
<i>HTTP Service</i>	68
<i>Import Mailboxes</i>	127
<i>Inbound Mail Policy</i>	99
<i>Mail Groups</i>	93
<i>Mailboxes</i>	87
<i>Mailing Lists</i>	92
<i>News Group</i>	141
<i>News Group Subscriptions</i>	140
<i>News Server</i>	143
<i>News-To-Mail Gateway</i>	146
<i>Post Master.</i>	85
<i>Scheduled Dial-Up</i>	62
<i>Scripts and Templates</i>	247
<i>Server Properties</i>	55
<i>Service Manager.</i>	222
<i>User Accounts</i>	123
<i>User Manager</i>	122
<i>Web Crawler</i>	182
<i>Web Crawlers.</i>	181
<i>Digests</i>	160
<i>Digital Signatures</i>	114
<i>Directory Visibility.</i>	84,183
<i>Distribution List</i>	154
<i>Domain Name Server</i>	
<i>See Domains</i>	
<i>Domains</i>	
<i>About</i>	10
<i>Aliases</i>	17
<i>DNS</i>	77
<i>Primary Domain</i>	18
<i>Dragging Mail Slots.</i>	43
<i>DUN</i>	
<i>See Dial Up Networking</i>	
<i>Dynamic IP Addresses</i>	314
<i>Dynamic Routing Table</i>	110

E

<i>Editing Web Pages</i>	185
------------------------------------	-----

E-Mail

<i>Addresses</i>	10
<i>Introduction</i>	70
<i>Message Envelopes</i>	71
<i>Message Headers</i>	72
<i>Smart Hosts.</i>	19,110,316
<i>Envelopes</i>	
<i>See Messages</i>	
<i>Event Logging</i>	128
<i>Event Reports</i>	131
<i>Expiry</i>	
<i>Messages</i>	89,200
<i>News Articles.</i>	141

F

<i>Filters</i>	105,119
<i>Finger</i>	
<i>Records</i>	52
<i>Service</i>	229
<i>Firewalls</i>	224
<i>Forums</i>	161
<i>Forwarding</i>	
<i>From a Mailbox</i>	88
<i>Undelivered Mail</i>	100
<i>Functions</i>	
<i>AddressOf()</i>	292
<i>AppendFile()</i>	272
<i>CallDLLFunction()</i>	308
<i>CJ()</i>	285
<i>CloseFile().</i>	273
<i>ConsoleMenuCommand().</i>	306
<i>CreateFile()</i>	272
<i>DateTimeToReal()</i>	289
<i>DateToReal()</i>	289
<i>DBDeleteRecord()</i>	296
<i>DBRead()</i>	295
<i>DBRecordExists()</i>	296
<i>DBWrite().</i>	295
<i>Define()</i>	270

<i>DeleteFile()</i>	273
<i>DeleteHeader()</i>	278
<i>Do()</i>	266
<i>EOF()</i>	273
<i>Execute()</i>	307
<i>ExecuteAsync()</i>	307
<i>For()</i>	268
<i>FormatDateTime()</i>	290
<i>GetDirectory()</i>	275
<i>GetSystemVar()</i>	305
<i>Header()</i>	278
<i>HostOf()</i>	292
<i>If()</i>	266
<i>Length()</i>	286
<i>ListCount()</i>	288
<i>ListItem()</i>	288
<i>LJ()</i>	285
<i>LoadHeader()</i>	283
<i>LoadMessage()</i>	283
<i>MsgAddLine()</i>	280
<i>MsgDeleteLine()</i>	280
<i>MsgGetLine()</i>	279
<i>MsgGetLineCount()</i>	280
<i>MsgGetRcpts()</i>	281
<i>MsgGetSender()</i>	280
<i>MsgInsertLine()</i>	279
<i>MsgSetRcpts()</i>	281
<i>NameOf()</i>	293
<i>OpenFile()</i>	271
<i>OriginalMessage()</i>	279
<i>Params()</i>	287
<i>PostMessage()</i>	281
<i>Read()</i>	274
<i>RealToDate()</i>	290
<i>RealToDateTime()</i>	290
<i>RealToTime()</i>	290
<i>Repeat()</i>	267
<i>RJ()</i>	285
<i>SameAddress()</i>	293
<i>SelectInMessage()</i>	277
<i>SelectOutMessage()</i>	278

<i>SetHeader()</i>	278
<i>SetSystemVar()</i>	305
<i>ShellExecute()</i>	308
<i>StrCmp()</i>	289
<i>SubStr()</i>	287
<i>TimeToReal()</i>	289
<i>UserOf()</i>	292
<i>While()</i>	267
<i>WildcardMatch()</i>	286
<i>Write()</i>	274

G

Gateways

<i>KA9Q</i>	230
<i>Mail-To-News</i>	146
<i>News-To-Mail</i>	144
<i>Pegasus</i>	231 - 232
Go Offline Command	67
Go Online Command	59

H

Home Pages	184
Hops	6
Host Assignments	111
HTML	176
HTTP	228
<i>GET Command</i>	187
<i>Overview</i>	177
<i>POST Command</i>	187
<i>PUT Command</i>	185
HyperText Markup Language	
<i>See HTML</i>	
HyperText Transport Protocol	
<i>See HTTP</i>	

I

Idle Timeout	63,67
Importing Mailboxes	126
Inbound Router	97,107
Installation Wizard	16
Internet	
<i>Connection</i>	53
Internet Service Providers	7,9,12,17
ISDN	57

K

KA9Q.	91,230
---------------	--------

L

Large Messages	107 - 108
Latency	111,142
Leased Line	55
List Commands	
<i>Members</i>	167
<i>Resume</i>	167
<i>Set</i>	166
<i>Status</i>	166
<i>Subscribe</i>	165
<i>Suspend</i>	167
<i>Unsubscribe</i>	166
List Mirroring	147
List View	43
Local News Groups	148
Local Router	98
Logging Method	129

M

Mail Barring.	101
Mail Exchange	

<i>About</i>	77
<i>MX Lookups</i>	77
<i>MX Resolution</i>	110,315
<i>Routing</i>	76
Mail Forwarding	88
Mail Gateways	81
Mail Gateways	90
Mail Groups	47
Mail Hops	6
Mail Routes	109
Mail Slots	
<i>Aliases</i>	201
<i>Archives</i>	91
<i>Auto Expiry</i>	89,200
<i>Introduction</i>	83
<i>Mailing Lists</i>	91
<i>The newsserver</i>	147
<i>Ownership</i>	125
<i>Postmaster</i>	48
<i>The mailserver</i>	162
<i>Visibility</i>	84
Mailbox Providers.	8
Mailboxes	87
Mailing Lists	
<i>Announcements</i>	161
<i>Control Messages</i>	163
<i>Digests</i>	160
<i>Distribution Lists</i>	154
<i>Even Distribution</i>	157
<i>Forums</i>	161
<i>Introduction</i>	153,160
<i>The mailserver</i>	162
<i>Maintainer</i>	157
<i>Mirroring</i>	147
<i>Moderating</i>	159
<i>Secure Lists</i>	164
<i>Subscribers</i>	158
<i>Subscription by mail</i>	155,161
Mailserver	49,162
Mail-To-News Gateways	146

Mailing Lists	
<i>Moderating</i>	195
Marquee Selection	44
Message Sorting	104
Messages	
<i>Envelopes</i>	71
<i>Expiry</i>	89
<i>Forwarding Mail</i>	88
<i>Header Fields</i>	72
<i>large messages</i>	107 - 108
<i>Mailtraq Message Format</i>	135
<i>MIME</i>	160
<i>Script Triggers</i>	248
<i>Templates</i>	168
Messaging	4
MIME	160
Modems	57
Moderating	
<i>See Mailing Lists</i>	
MX Resolution	76,110

N

Network Latency	111,142
Network Services	
<i>Firewalls</i>	224
<i>Introduction</i>	221
<i>Remote Logging</i>	229
News	
<i>Article Expiry</i>	141
<i>Authentication</i>	150
<i>Enabling During a Schedule</i>	64
<i>GROUP Type Collection</i>	144
<i>Groups</i>	140
<i>Internet News</i>	137
<i>Local News Groups</i>	148
<i>NEWNEWS Command</i>	142
<i>Posting</i>	151
<i>Servers</i>	142
<i>XHDR Command</i>	144

<i>XOVER Command</i>	144
Newsserver	49,147
News-To-Mail Gateways	144
NNTP	150,227

O

On-Disk Event Logging	129
On-line	58
Outbox	49,100
Outgoing Mail	109,111
Outgoing News	151

P

Parallel Mail Delivery	111
POP3	21,46
<i>Collection From</i>	115
<i>Enabling During a Schedule</i>	64
<i>ENV Command</i>	117
<i>Overview</i>	78
<i>Preventing Duplicate Downloads</i>	113
<i>Preview Window</i>	120
<i>Routing</i>	116
<i>Service</i>	114,227
Post Master	85
Postmaster Mailbox	48
Preview Window	120
Proxy Servers	178
Proxy Tunnels	228
Public Archives	169

Q

Querying Connections	66
Queue Mode	
<i>See KA9Q</i>	

R

RAS

See Remote Access Service

Re-Addressing Mail	100
Registration Details	16
Registry	134
Remote Access Service.	13
<i>Dial Up Networking</i>	55
Remote Logging Service	229
Remote Mailboxes	21,46,114
Reports	
<i>Accounting</i>	93
<i>Delivery Reports</i>	100
<i>Event Reports</i>	131
Return Path	102
Returning Undelivered Mail	99
Re-Writing	102
Routing	
<i>Advanced Routing Table</i>	110
<i>Inbound Router</i>	97,107
<i>Local Router</i>	98
<i>Outbound Router</i>	109,111
<i>POP3 Collection</i>	116
<i>Smart Hosts</i>	110
Run Script Command.	246

S

Schedule Now Command	67 - 68
Schedule Options	66
Scheduled Dial-up	60
Scripting	
<i>Automated Scripts</i>	246
<i>Constructs</i>	239
<i>Error Handling</i>	242
<i>Expressions</i>	239
<i>Functions</i>	237
<i>Introduction</i>	235

<i>Lists</i>	243
<i>Operators</i>	238
<i>Run Script Command</i>	246
<i>Strings</i>	244
<i>Variables</i>	238
Secure Password Authentication	114
Secure Sockets Layer (SSL)	114
Security Timeout	190
SGML	176
Smart Hosts	19,110,316
SMTP	7,9
<i>Overview</i>	74
<i>Service</i>	112,226
<i>Transactions</i>	75
Sorting	104
Static IP Addresses	317
Static Routing Table	110
Subscription By Mail	155
Subscription Commands	
<i>See List Commands</i>	
Subscriptions	50
Synchronised Online Times	56
System Tray Icon	40

T

TAPI	13
TCP/IP	
<i>About</i>	317
<i>Ports</i>	74,316
<i>Proxy Tunnel</i>	228
Telephony	
<i>See TAPI</i>	
Templates	
<i>Introduction</i>	253
<i>Mailing Lists and Templates</i>	168
<i>Sub-classing</i>	259
Tree View	43
Trigger Messages	248

U

Undelivered Mail	99
Universal Resource Locators	
<i>See URL</i>	
URL	177
URL-Encoded Format	186
USENET.	138
User Information Protocol	229
Users	
<i>Accounts</i>	122
<i>Mail Slot Ownership</i>	125
<i>Web Administrator Access</i>	191

V

View Event Log.	130
View Record Command	52
Virtual Network	56
Visibility	84,183

W

Warn on Missed Schedules	67
Web Administrator	

<i>Active Connections</i>	217
<i>Directory</i>	192
<i>Introduction</i>	189
<i>Mailing Lists</i>	203
<i>Moderating Messages</i>	195
<i>Reading Mail.</i>	194
<i>Reading News</i>	211
<i>Security Timeout</i>	190
<i>System Administration</i>	214
<i>User Access</i>	191
Web Browser	177
Web Cache	179
Web Crawlers	
<i>Overview</i>	180
<i>Scan Depth</i>	183
Web Proxy	179
Web Server	183
Wildcards	102
Windows Clipboard	131
Windows Explorer	40
Windows Registry	134
World Wide Web	175

X

X-Moderate-For	159
--------------------------	-----