

ARTWARE

presents to you:

timEd

v1.01

(c) 1992,93,94 Gerard van Essen (2:281/527)

A fast message editor for Squish, JAM, *.MSG & Hudson.

timEd version : 1.01
Document last updated : October 21 1994

Note: Features marked with {+} are only available in registered versions of timEd.

Programming code credits:

! timEd uses the Squish MSGAPI by Scott Dudley.
! Squish is a trademark of Scott J. Dudley
! timEd uses Thomas Wagner's EXEC library to swap timEd out of
memory.
! timEd's screen writing routines are partly based on code from
jim nutt and Erik Vanriper (DOS version).
! timEd uses Mark Potter's DVWARE routines (DOS version).
! JAM(mbp) - Copyright 1993 Joaquim Homrighausen, Andrew Milner,
Mats Birch, Mats Wallin. ALL RIGHTS RESERVED.

Documentation credits:

Documentation written by Gerard van Essen.
Mac Chou designed the artwork for the title page.
Jelle Boot wrote the 'Quick installation' part.

Contents.

1.	Introduction.	5
1.1	What is timEd?	5
1.2	A highlight of timEd's features.	6
1.3	History	7
1.4	System requirements	8
2	Legal stuff	9
2.1	Copyright, license and disclaimer.	9
3.	Quick installation	11
4.	Command line parameters & errorlevels.	12
4.1	Command line parameters.	12
4.2	TimEd exit errorlevels	12
5.	A guided tour of some of timEd's features	13
5.1	The Area Selection Screen	13
5.2	The message reading screen	14
5.3	Entering a message	17
5.4	Netmail messages	19
5.5	The List mode	19
5.6	The Find Screen	20
6.	TimEd command reference	22
6.1	The Area Selection Screen.	23
6.2	The Message Reading Screen.	25
6.3	The List Mode.	35
6.4	TimEd's Find feature.	35
7.	TimEd keyboard reference	41
8.	Effective use of timEd: some features explained	45
8.1	Creating netmail messages.	45
8.2	AKA matching.	48
8.3	TimEd's Carbon Copy (CC:) feature.	49
8.4	TimEd's crossposting feature.	50
8.5	Message attributes	51
8.6	Using the internal editor	55
8.7	Using an external editor	58
8.8	Reading messages 'gated' from Usenet/Internet	60
9.	Message base formats explained	61
9.1	*.MSG	61
9.2	Hudson	62
9.3	Squish	63
9.4	JAM	65
10.	timEd configuration reference	67

11.	Miscellaneous notes.	97
11.1	Multitasking considerations.	97
11.2	Some things to remember.	97
11.3	Technical notes.	98
11.4	TimEd source code	98
12.	Frequently asked questions	99
13.	Thanks to..	101
14.	Contacting the author.	102

1. Introduction.

1.1 What is timEd?

TimEd is an electronic mail editor for FidoNet.

FidoNet is a network of computer hobbyists, which makes it possible to exchange both private and public messages between systems connected to this network, using the public telephone network as its transmission medium.

In-depth discussion of FidoNet is not appropriate here, there are other sources for that information.

In order to send and receive messages, there are three main tasks to be fulfilled:

- 1) Reading and generating messages.
- 2) Packing and unpacking messages into/from standard mailpackets that can be easily exchanged.
- 3) Transmitting and receiving mailpackets.

Most of the time, these three separate tasks are fulfilled by three different programs.

Programs that allow the user to read and write messages are called message editors. TimEd is a message editor, that will allow you to comfortably read and write messages.

Programs that will pack and unpack messages are usually called echomail processors (or 'tossers'). Some examples of echomail processors that are used in FidoNet: Squishmail, Fastecho, GEcho, Fmail.

Programs that transmit and receive mailpackets are called mailers. Some examples of FidoNet compatible mailers: FrontDoor, BinkleyTerm, The Box, Portal of Power.

Although the separate tasks are usually fulfilled by different programs, these programs have to work closely together: the echomail processor must unpack the messages into a format that can be read by the message editor, just as newly generated messages must be generated in a format that can be understood by the echomail processor.

There are several different ways to reach the same goal, and hence, as you might have guessed, there are several different formats available to store messages on a system. Each of these formats has strong and weak points.

The collection of messages stored on the system is usually referred to as 'the message base' and the format in which they are stored is called 'the message base format'.

In order to be of use to as many people as possible, timEd fully supports no less than 4 different message base formats. This covers all popular formats used in FidoNet, as supported by the most often used echomail processors.

1.2 A highlight of timEd's features.

- * TimEd was designed to offer speed and ease of use: the program is fast and takes little of your precious disk space. It is also quite easy to install.
- * Support for all major message base formats that are used in FidoNet: Squish, JAM, Hudson and *.MSG
- * Direct support for nodelist indexes used by FrontDoor and Version 7 nodelists: no need for large, extra nodelist indexes, taking up lots of disk space. TimEd will just use the nodelist that is already present on your system!
- * Support for most popular echomail processors: timEd can read the echomail areas from area configuration files generated by Squish, Fastecho, GEcho, Fmail, Imail, Watergate and xMail. In addition, it will read standard AREAS.BBS format files, that can be generated by most other echomail processors.
- * An internal, 'quotesmart' editor, with automatic wordwrapping (including intelligent reformatting of quotes) and useful block commands.
- * Optional use of an external editor. If you don't like the internal editor, you can still use the ASCII editor of your choice.
- * Useful list mode, with the option to tag several messages and move, copy, delete or file all tagged messages with one command.
- * Comprehensive 'find' function. This will let you search for messages containing keywords you specify in the header, kludges or the body of the message. Search options include 'AND' searches, 'OR' searches, 'NOT' searches, case sensitive searches and searching for 'whole words only'.

The function can be used to scan the current area only, all areas or tagged areas, and to scan all messages or only new messages. Searching can be forward or backward, and the search direction can even be switched during a search.

Found messages can be read (the registered version will even highlight the lines containing any of the strings you are looking for!), tagged, written to a file, copied, moved or deleted.

- * Built-in, fast personal mail scanner. This will search all message areas for messages addressed to you. (This feature is only available to registered users).
- * Automatic file request feature. TimEd can extract valid filenames that are found in the message text and let you launch a file request, after you selected some files from a picklist with the filenames that were found.
- * Support for messages that were gated from Usenet, allowing you to send private 'mail' without having to worry about the complicated usenet addresses: timEd will put them in the right place for you (FSC-0035).
- * DOS & OS/2 versions available.
- * Lots more :-)

1.3 History

TimEd is one of those programs that 'were never meant to be'.

When Scott Dudley released Maximus 2.00 and Squish 1.00, I immediately switched (from Qmail, *.MSG) to the much more advanced Squish message base format. However, initially there were no message editors available, so I had to use Maximus itself to read the mail. Although this went reasonably well, it wasn't really fast and I missed some features ('netmail reply' being the most important one).

When Scott released the MSGAPI (and I had just bought my Turbo C compiler :-) I played with it a bit, and decided to make a message viewer (yes, viewer only!).

I got carried away.

Later several message editors for Squish were released, but I always found reasons not to switch (too large, too slow, too buggy, ugly, whatever :-).

I continued working on timEd, other people showed interest, got their copies and some of them even liked timEd!

So.... I decided I would try to make a stable version, that could be released 'to the public' to check out the reactions (if any). Of course, a program is never ready (or without bugs), so I delayed it several times.

Nevertheless, I released a first 'wide beta' version to the public in november '92 as timEd/B5.

Still unsure about the need for yet another message editor, I was pleasantly surprised by the amount of reactions I got. Most of them were quite positive as well, which made me motivated to continue to work on the program.

Several other beta versions were released, until we reached beta 9.

And then there was this deja-vu thing: a new message base format was introduced, called JAM. And an accompanying API was released as well!

So there it was, sitting on my harddisk: the specifications of a new and exciting message base format, with accompanying sample code for me to experiment.

Like a beautiful woman, I couldn't get my mind off it. How could I resist?

So, being one of the first, timEd also supported JAM shortly after that. This introduced timEd to a new group of users, many of whom were still using the Hudson base. With requests for Hudson support overflowing my mailbox, I took the plunge and implemented support for the Hudson message base as well..

At the same time, the internal editor was finally ready as well, adding the last feature that was (in my opinion) still missing, but should be present in a decent message editor.

All of this paved the way to the first official non-beta release of timEd: v1.00.

But this is it! This version has all the features I wanted my editor to have, in order to work comfortably. I am not very fond of programs that are full of 'bells and whistles', and you won't find many gee-whiz-bang features in timEd (although it does have some special things, like a built-in personal mail scanner and a nice 'Find' feature).

If you are looking for soundblaster support, random origins, taglines and configurable beep-lengths, you better go and find yourself another program.

1.4 System requirements

- * An IBM compatible computer.
- * MS-DOS 3.3 or later, or OS/2 2.0 or later.
- * About 350 Kb of available memory.

2 Legal stuff

2.1 Copyright, license and disclaimer.

* "timEd" refers to the executables and documentation in the original distribution archive. TimEd is copyrighted material by Gerard van Essen. It may only be used in agreement with the conditions set out in this license agreement.

* You are entitled and encouraged to copy and distribute timEd, provided you do not change the contents of the timEd archive or the program itself, and no money or any other compensation is asked or accepted for timEd (without written permission from the author).
Distribution of modified or incomplete archives is prohibited.

* Although care has been taken to write and test a program that does what this document states, the program is provided as is, without warranty or guarantee of any kind, either expressed or implied, as to the quality or performance of this program, except that it will occupy disk space.

* The author, Gerard van Essen, will not be held liable to you or anyone for (but not limited to) any direct, indirect, incidental or consequential damages, including any lost profits, lost savings which may result from the use or inability to use this program.

Gerard van Essen is in no way obligated to provide future versions of, or support for this software.

Your use of the program constitutes your agreement to this license and disclaimer and your release of the author from any form of liability or litigation.

* COMMERCIAL use of the program: you are a commercial user if you make a profit from running timEd, or if you use timEd in a commercial environment (ie. business, governmental organization, association, school, foundation, or any other form of juridical person). In case of doubt, contact the author.

* For NON-COMMERCIAL use, it is REQUIRED that you register the program after an evaluation period of 30 days. This period should be sufficient to decide whether you want to continue using the program or not.

I would very much appreciate registration, and it will stimulate me to keep on developing the program.

If you really cannot afford to register (mainly a concern to 'eastern block' countries, the former USSR etc.), permission is hereby granted to use an unregistered version.

Remember, if you register, you are paying for something that you already have! Registration does not mean you can force me to implement new features that you like.

- * For COMMERCIAL use of the program, you should always register the program. You are, however, granted an evaluation period of 21 days. After these 21 days, you must either register the program, or stop using it. Using an unregistered version of timEd in a commercial environment for more than 21 days is prohibited!
- * You only have to register once. Your registration will also be valid for all future releases of timEd.
- * Registration is valid for all versions of timEd. At this moment, there are DOS and OS/2 versions of the program available. Registering timEd entitles you to run the DOS version, the OS/2 version, or both :-)
- * A registration is PERSONAL. It cannot be transferred to other parties. It could, however, be used in different places (but: by the same PERSON). Think of it as a book in this regard: you can take the book from your home to the office (and read it there).
- * Versions of timEd prior to version beta 10 (timEd/B10) were freeware and could not be registered. You may continue to use THESE OLDER VERSIONS without registration, even in a commercial environment.
- * The author reserves the right to change this license without prior notice, for newer versions of the program.

3. Quick installation

Assuming you've just downloaded the timEd archive from your favourite BBS the first thing to do is decide where to put the executable, helpfile and config file. You could give timEd a directory of its own. However, due to the small size of the program most users prefer to place timEd in the same directory where their mail processor lives.

Next action is to edit timed.cfg to suit your needs and your system. The file that is included in the archive is heavily commented and generally just changing the various paths to reflect your own directories and of course inserting your own address already brings you halfway to a working system.

The only part that requires some effort is the Area Definition section. TimEd will normally read all your EchoArea definitions from your mail processor's config file.

As soon as you're in doubt about a certain keyword, refer to the section called 'timEd configuration reference' in this manual. This is an alphabetic listing of all keywords that can be used in timed.cfg, with an explanation of their use.

Just follow the examples provided and when you are finished you're off to the races. Fire up timEd and enjoy.

4. Command line parameters & errorlevels.

4.1 Command line parameters.

TimEd knows 2 command line parameters:

-c

Specify the directory (yes, the DIRECTORY, not a config file!) where the config file (and helpfile) can be found, like:

```
timed.exe -cC:\Bbs\timed\
```

Usually, you do not need this. TimEd will look for timed.cfg in the current directory by default, where it can usually find it..

And the other command line parameter is -p, to let timed pause at the intro screen (so you can check what config files were parsed by timed). Like:

```
timed.exe -p
```

Of course, you can combine both switches:

```
timed.exe -p -cC:\Bbs\timed
```

4.2 TimEd exit errorlevels

TimEd can exit with several useful errorlevels:

```
No mail entered      : 0
Netmail entered      : 1
Echomail entered     : 2
Local mail entered   : 4
```

If more than one type is entered the numbers are added:

```
Net + Echo           : 3
Net + Local           : 5
Echo + Local          : 6
Net + Echo + Local    : 7
```

```
Error: 254
```

You can trap these errorlevels in a batchfile, and scan echomail or pack netmail (or both) based on the errorlevel.

5. A guided tour of some of timEd's features

This section will show new users some of timEd's features, as an introduction. If you are new to electronic mail in general, you may find this useful.

If you are an experienced 'electronic mail reader', you will find timEd easy to use, because it works like most other electronic mail editors. Some keys may be a bit different, but usually not.

This section is a 'tour', NOT an in-depth discussion of all the 'ins and outs'! There's much more to know than what is presented here!

Pressing F1 will usually give a helpscreen, listing the commands available in the section of the program that you are currently in. If F1 doesn't work, you should check for the presence of TIMED.HLP in the timEd directory.

Let's assume you have already edited timEd.cfg to suit your needs, and that there are some areas (at least a netmail area and one echomail area) installed in there.

If there are not, please add some 'fake' areas to the config, just to be able to read and enter some messages..

5.1 The Area Selection Screen

After starting up timEd, and going through the intro screen, you will be presented with a list of all installed message areas. The list will look like this:

Area description:		New	Last
Netmail Area	NETMAIL	0	47
Netmail squish	net2	0	589
Netmail squish2	net3	0	2
Netmail (rcvd)	rcvd	0	0
Netmail (sent)	sent	0	0
Netmail JAM	netjam	0	193
Netmail JAM2	netjam2	0	149
Netmail HMB	nethmb	0	2
contrast	contrast	0	7
os2.028	os2.028	* 11	110
artware	artware	0	155
artware.closed	artware.closed	0	0
contrast.news	contrast.news	0	780
tub	tub	* 4	216
points.133	points.133	0	146
fdecho.028	fdecho.028	0	48
net_dev.028	net_dev.028	* 5	144

One of the areas (the first one, when you first start timEd up) will be 'highlighted'. You can move around through the areas with the cursor keys, page-up and down etc.

On every line, the first part is the description of the area, the second part is the official 'areatag' of that area. Then there are two numbers: the first one is the number of new, unread messages that are present in the area, the second one is the highest message number in that area. Every area that contains new messages will be marked with a star ('*') in front of the numbers.

At startup, you might see '- -' instead of the numbers. This means the area was not yet scanned for new mail. You can scan for a new mail by pressing ALT-S (Scan).

You can now move around to find the area you want to read. When you found it, press <enter>.

Tip: to quickly find an area, you can also start typing the 'areatag'. TimEd will quickly jump to the first area with an areatag that matches the letters you typed (even if the string you are typing is hidden somewhere in the middle of the name, like the string 'os2' in comp.os.os2.programming!).

To exit timEd, press ALT-X (don't do that now, we want to finish our tour!).

Remember, F1 gives you a help screen if you forgot the commands!

5.2 The message reading screen

After you selected an area to read, you will be presented with the lastread message of the area you entered. If there are no messages present in that area, a message indicating that unfortunate situation is shown (in that case, proceed with 'entering a message' and then come back here..).

In netmail areas, this is different, addresses are always present there, and don't have to be deduced from the text. In addition, netmail also has a destination address, that is shown below the origin address:

= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -

```
Date : Apr 08 '94, 18:50          Pvt Loc
From : Gerard van Essen          2:281/527.1
To   : Evin Mulron              1:138/211.0
Subj : Testing..
```

= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -

This netmail message is addressed to Evin Mulron, and his address is 1:138/211. As you can see, this message also has the 'pvt' attribute (private).

Now let's return to our sample echomail message, and look at the body again:

= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -

Hello All!

 This is just a test!

Groetjes, === Art ===

```
--- timEd 1.00+
* Origin: CONTRAST | TSARTNOC (2:281/527.1)
```

= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -

You can see the text of the message. Then there is a line with '---timEd 1.00+', called the tearline. This line should show nothing at all (so just '---'), or else the name of the message editor or echomail processor the sender was using.

Then there is a line with ' * Origin: (<address>)', that shows some text and ends with the address of the person who sent the message (which is the place where timEd got the address that was listed in the header of this echomail message).

Finally, the bottom line shows:

```
= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -
```

```
Area: contrast          100%          Current: 9, high: 9 | 318K
```

```
= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -
```

First the current area is shown. Then a percentage, showing how much of the current message was already shown to you, the number of the current message, the number of the highest message, and finally the amount of memory that is free (only in the DOS version).

This is the screen you will be looking at most of the time. You can use the cursor keys (up and down), home, end, page up & down to move the message body so you can read all of it.

You can use cursor right and left to move to the next or previous message. If you entered a message area with lots of messages, you can now move around in the message area to read all of them.

To quickly move to the first message in an area, press ctrl-<home>, to go to the last one, press ctrl-<end>.

To go back to the Area Selection Screen, press <ESC>. Press <enter> on any area to enter an area again.

Provided you have already linked up some areas with your uplink, and mail is already flowing, this basic knowledge is already enough to keep you busy for several hours a day. But there's more...

There's a lot you can do with a message. The F1 help screen will reveal many more functions: writing a message to a file ('W'), Printing it ('P'), deleting it with , move or copy it to another area with 'M' etc. Just try it!

Some other nice things:

Press 'I'. You will be presented with more info about the message, and about the message area you are currently in.

Press 'K' to see hidden control info in the message (if there is any).

Press 'J' to temporarily jump to DOS or OS/2 to do something else for a while.

5.3 Entering a message

Reading is good, writing is better. Now we know how to read a message, we also want to write one!

In order to do this, go into an echomail area, and press <INS>.

You will now be provided with a fresh message header, with your name already filled in, for you to fill out:

= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -

Date : Apr 08 '94, 19:25
From : Gerard van Essen
To :
Subj :
Loc 2:281/527.1

= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -

You see the cursor at the TO: field. Type a name if you want to send the message to a particular person, or just 'All'. Press <enter> and edit the subject. Press <enter> again, and once again (you will be presented with an 'attributes' screen, but we press <enter> and leave everything at the default values).

What happens next depends on your setup. It is time to enter the message body, so either the external editor that you installed is started up, or timEd's internal editor kicks in. Lets' assume you use the internal editor:

= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -

```
Area: contrast      To: All (Test)
Hello All!
```

Groetjes, === Art ===

* Origin: CONTRAST | TSARTNOC (2:281/527.1)

= - = - = - = - = - = - = - = - = - = - = - = - = - = - = - = -

We see timEd did a lot of work for us (depending on what you specified in timEd.cfg).

The top line shows the area, and the person(s) the message is addressed to ('all') and the subject between brackets ('test').

TimEd already entered 'Hello All' at the top for us, and ended with (Dutch) greetings.
It generated an Origin line as well, that is already filled in.

We can now easily finish the message text to our liking. Several commands are available (see section on the internal editor, or press F1 for help).

Once we're ready, we press ALT-S to save the message. TimEd will do some work for you, generating the actual message, and we will return at the Message Reading screen.

Press <ctrl-end> to move to the last message in the area, which is (of course) the message we just created! There, your first message generated with timEd!

You know how to enter a new message now, but usually you want to enter a reply to another messages. To do this, press ALT-R (or just 'R'). Once again, you will have to edit the header, but timEd will already have filled in some default values, that may save you the typing if they are what you want. You can then simply press <enter>.

When you're ready to edit the message body, you will see that the text of the message you are replying to is already present in the message body, in 'quoted' format: it is preceded by strings like: 'GvE> ' (the initials of the writer of the message and a '>' token to indicate you are quoting someone else).

Delete all the lines that are unimportant (or not needed to remind others what you are actually replying to) and insert your reaction to the message.

In general: quote only what is really needed. Less quoting is better, you're only repeating what was said before!

A very fast way to reply is to press 'T'. This will not let you edit the header at all, but automatically inserts the default values and immediately puts you in the message editor to edit the body.

Once again, pressing ALT-S will save the text you typed, and generate a message with it.

5.4 Netmail messages

As was mentioned before, netmail is somewhat different from echomail, because in netmail real electronic addresses are present (and necessary). This means, that when you edit a message header in the netmail area, you must also know the address of the person you are sending a message to.

TimEd can help you there, with nodelist lookups (you only enter the name, and timEd searches the nodelist for the actual address). Look at the section 'entering netmail messages' later in this manual for more info.

5.5 The List mode

Apart from looking at messages one at a time, you can also get an overview of the messages in the area using the List mode. While looking at the message reading screen, press 'L'.

You will then be presented with a list of message headers, looking like this:

1	Sjaak Paridaens	Gerard van Essen	Printen
2	Gerard van Essen	Sjaak Paridaens	Printen
3	Andre Boogaards	Gerard van Essen	Netwerk, rebooten en
4	Andre Boogaards	Gerard van Essen	The Sequel...
5	Gerard van Essen	Andre Boogaards	Netwerk, rebooten en
6	Dick Pluim	Gerard van Essen	nog een bug?
7	Gerard van Essen	Dick Pluim	nog een bug?
8	Andre Boogaards	Gerard van Essen	Netmgr masking idee
9	Gerard van Essen	All	Testing..

This looks a lot like the Area Selection Screen, but it shows message headers, not message areas. You can move around with cursor keys etc. here as well, just like the Area Selection Screen. It gives you a nice view of the area. As soon as you see a header that interests you, press <enter> on that header, and you will drop to the Message Reading Screen again, to read the entire message.

Now press S while you are in the List mode. You will see that the TO: fields are not shown anymore, but the 'Subject' lines get a lot more space. This is mainly for people who read messages gated from Usenet. Those messages usually all have 'all' as the to: field, so that space can be used for the subject line! You can also get into this mode by pressing 'B' (Broad List) while you are at the Message Reading Screen.

You can also tag messages from the List screen, by pressing <space>. (Also by pressing '+' to specify a range, and more, see help screen). Use this to tag a lot of messages, and then Move, Copy, Delete, Write to file or Print a whole bunch of messages with the touch of a key!

Once again: the F1 help screen will give you all the commands!

5.6 The Find Screen

The last stop in this tour is one of the nicest functions in timEd: the Find feature. It will let you search for messages that have a certain keyword in them, in either the header (the From field, the TO: field or the subject) or the message body.

TimEd can search all areas, or just the current area. It can search all messages in the area, or just the new, unread messages.

You can activate the Find feature by pressing 'F'. From what you've seen upto now, you know that the first thing to do is pressing F1 to see some help on the keys to use. It will be pretty easy after that. Fill in the 'find form' (press <ins> to add a new item) and then press ctrl-<enter> to start the search. More info about the Find feature can be found in a special section about it later in this manual.

Also check out the file request feature, which is described later in the manual as well!

This ends our introductory tour. The key to success in timEd is called 'F1'. Remember that, and you'll be on your way! Many more useful hints and tips can be found in the rest of this manual. If you don't read it in its entirety, you may miss out on some pretty useful things. That will be your loss.

Good luck!

6. TimEd command reference

This section explains just about every command available in timEd. It usually just tells you how it works, but often not why it works like that, or how you can use it best. If you want to know more about the commands, you might find hints and tips in the chapter called 'Effective use of timEd: some features explained'.

First, we start off with a short description of the string editing function, that is available from several places in timEd.

▪ String editing in timEd

In timEd, there are several functions that let you edit strings. Some examples include generating a message header (TO: field, address, subject) and the 'write message to file' function, where you have to enter a filename.

All string editing functions use the same routine, where the following keys are active:

Right arrow, left arrow : move left and right in string.

<end>, <home> : go to the beginning or the end of the string.

<ctrl-left>, <ctrl-right> : jump to previous or next word.

<ctrl-end> : delete string from cursor position to end.

Delete and backspace work as expected.
Press <INS> to toggle 'insert mode'.

And, very useful:

CTRL-Y : Delete entire string.

You can also influence timEd's behaviour with the Jumpy_Edit keyword. See the section listing all keywords for more info.

ALT-J will let you shell to the OS when editing any string.

Whenever you edited a string, you should press <enter> to make the actual change. If you changed your mind, you could press <ESC>. However, sometimes you are editing a set of fields (like editing a message header), where pressing <ESC> would abort the entire function. In that case you can also press <TAB> (which means: jump to next field, don't change current contents of string).

Now, these are the commands you can use in timEd:

6.1 The Area Selection Screen.

After starting timEd up, you will be presented with the Area Selection Screen: a list of all message areas found in your configuration.

Use the cursor keys (up and down), <page up>, <page down>, <home> and <end> to move around.

A quick way to find an area is 'speedsearch'. If you know the official areatag of the area you want to read, start typing that tag. TimEd will try to find the first areatag matching the characters you typed. The characters will also appear at the top of the screen.

If there is no areatag that starts with the letters you type, timEd will try to find the string you typed as a SUBstring somewhere within the areaname.

For example, if you have the following areas:

```
comp.os.os2.programmer
comp.os.os2.networking
comp.os.os2.advocacy
```

and type 'advoc', timEd will find no areatag that starts with 'advoc', but it WILL find one that has 'advoc' as a substring! So it will jump to the third area.

The 'speedsearch string' will disappear (and be cleared) when you press a 'movement key' (like 'cursor up').

If you press CTRL-L, timEd will try to find a next match for the speedsearch string (this may come in handy if you have an area called OS2, but also one called OS2.028, and they are not close to each other in the arealist).

To start reading messages in an area, press <ENTER> or the right arrow key. TimEd will then enter that area and display the last read message.

On the area selection screen, the following keys are also active:

ALT-S : scan areas for new mail. This will determine the number of messages present in every area, and the number of new (unread) messages. This function will scan all areas that have not been scanned before.

ALT-T : scan areas for new mail. This will determine the number of messages present in every area, and the number of new (unread) messages. This function will scan all areas, even if they were scanned before. Great if new mail was tossed in the background (on a multitasking system or network).

ALT-P : {+} personal mail scan. This will scan all areas for mail addressed TO: you or your alias (as defined in timed.cfg).

While viewing a found message, nearly all 'normal' functions are available to you (like R)eply, T)urbo reply, P)rint, W)rite to file etc).

Some keys (like cursor right and left) continue the search. You can interrupt the search at any time by pressing <ESC>.

ALT-X : Exit timEd. If specified in timed.cfg, an 'echotoss.log' file will be written at this point.

ALT-L : Enter the area, and go directly into the 'List mode', showing a list of message headers.

ALT-J : Jump to DOS.

<space> : Tag message area (see section about this below).

ALT-R : Read 'tagset' (for tagged areas, read section below).

ALT-S : Save 'tagset' (for tagged areas, read section below).

ALT-M : Select a 'view' on the arealist (all, new, tagged etc). See section below.

F1 : Show a helpscreen with available functions.

With <space>, you can tag (mark) areas. A collection of marked areas is called a 'tagset', and can be saved to disk using ALT-S. A tagset that was saved to disk, can later be read from disk again by using ALT-R.

Each set is identified with a number (the first one is 0, last one is 9), so you can save up to 10 sets. Tagset number 0 is the default, and will be automatically loaded from disk when timEd starts up.

Together with the ALT-M function (select 'view' on the arealist), this gives you the opportunity to only see and work on a selection of areas instead of the entire message base.

The following views are available:

- All areas.

This will show all areas found in the configuration file(s).

- Tagged areas.

This will restrict your view to the areas you tagged, others will not be shown (or scanned etc.). This is mainly useful for SysOps who have many areas they never intend to read anyway (but want to be able to

access them once in a while in order to check things). You will only see the areas you are interested in at that moment, and other areas will not be scanned either, which will save you a lot of time.

- Areas with new mail.

This will show only areas that contain new, unread messages. This is an easy way to read all new messages: the number of areas that is shown will decrease (an area you read will have no new messages anymore and therefore disappear from the arealist) until you have no more unread mail!

These two modes (tagged & new messages) can also be combined:

- Tagged areas with new mail.

This will show only areas that are tagged *and* have new mail.

The mode at startup can be set with a keyword: Startup_Mode (values: All, New, Tagged or NewTagged).

Together these function give you the ability to usually work on a subset of your areas and easily find areas with new mail, but switch to a full view with a keypress.

6.2 The Message Reading Screen.

When you enter an area, the lastread message will be shown.

You can scroll the message body (if it doesn't fit on one screen entirely) using cursor keys (up and down) and <page up>, <page down>, <home> and <end>.

<ESC> or ALT-A will take you back to the area selection screen.

Moving around in a message area.

Press the right arrow key to go to the next message, the left arrow key to go to the previous message.

<ENTER> will also move to the next message (unless you have 'EnterMarks Yes' in your timEd.cfg - in that case it will mark (tag) the current message and space can then be used to move a page down in the message (if there is more text to show) or to move to the next message (if you are already at the bottom of the text)).

<page up> and <page down> will move one page up or down in the message.

Ctrl-<END> will take you to the last message in the area.

Ctrl-<HOME> will take you to the first message in the area.

Type a number to go to a specific message. A small box will pop up, where you can edit the number of the message to jump to.

Ctrl-<up> will move to the next marked ('tagged') message in that area.

Ctrl-<down> will move to the previous marked ('tagged') message in that area.

Ctrl <left> will go to the 'original' message, i.e. the message that the current message is a reply to. If you are reading a reply and don't know anymore what the original message was about, use this.

Ctrl <right> will go to the reply to this message (if there are any replies already). If there is more than one reply, a list will be shown (with the name of the sender of the reply) so you can choose one.

The availability of replies and 'the original' is shown at the top of the message screen. An example:

--

```
Date : 20-10-'92, 21:38      17◀▶44
From : Jan Terpstra                2:280/216.0
To   : Hans Boelens
Subj : Negeer dit!
```

--

Here you can see, that this message is a reply to message 17, and that there is a reply to this message, number 44. If there would have been more replies (and you used a reply linker that is capable of specifying 'multiple uplinks', a feature that is only available for Squish style message bases) there would have been more numbers after the '44'.

For JAM style areas:

In addition to the 'normal' links, you may see a number between brackets, like this: (25). That means that there is another reply to the original message (it can only show up if you are reading a reply to a certain message). Pressing ALT-right or ALT+ will let you jump to that message.

An example:

You are reading message #5, it shows (at the top):

▶25 30

That would mean there are two replies to this message, number 25 and number 30.

You press ctrl-right (to get a pop-up window, displaying who wrote both replies), and choose the first reply (number 25).

This message (number 25) may show:

5◀▶50 (30)

This means that this is a reply to message 5, that there is a reply to this message (the reply is number 50), and that there is another reply to the original message (the original message was number 5), and that the number of that other reply is number 30 (which is exactly what message #5 showed us).

Confused? You should be!

Simple: if there is more than one reply to a message, you can press ctrl-right, go to the first reply, and then keep on pressing ALT-right (or ALT-+) to read all other replies as well. When you reached the end of all replies, you press ctrl-left to go back to the original message again..

Remeber, this only applies to JAM style areas!

ALT-E, E, <ins> : Enter message

This will let you create a new message in the current area. TimEd will first let you edit the header, and will then spawn the editor, where you can type in the message text. Save the text and exit the editor. You will then come back in timEd, that will read your message and save it.

See also the section 'Creating netmail and file attaches'.

ALT-R, R, ALT-Q, Q : Reply to message

This will let you write a reply to the message that is currently shown on the screen. TimEd will let you edit the header (press <ENTER> to accept the defaults) and quote the entire message. Then the editor will be spawned..

ALT-N, N : Reply other area

Create a reply, but select another area to put the reply in. This will first take you to the Area Selection Screen, where you can choose the area.

ALT-O, O : Reply 'followup'

This will let you quote the message text and add your comments, but will not address the message to the writer of that message, but to the person to whom the current message is addressed to.

So, when someone writes a message to John, and you use ALT-O on that message, it will be quoted and also addressed to John. You can then add "I want that too!", or something similar :-)

ALT-T, T : Reply, accept defaults

This is the equivalent of ALT-R, but you will go directly to the editor, accepting the defaults for the TO: and Subj: fields. This is a fast way to reply, for lazy people with little time :-)

CTRL-B: Bounce-reply

This is only available in netmail. It will generate a reply, but not quote the message. Instead, it returns the original message text (including a copy of the message header).

It will dump you in the message editor, so you can add the reason for your bounce-action.

ALT-C, C : Change message

This will let you change the message (header + body).

CTRL-H : Change header

This will you change the header of a message only.

ALT-S, S : Change message attributes

This will let you edit the message's attributes (like private, crash etc.).

CTRL-U : Unreceive message.

This turns *off* the 'received' bit. Be sure *not* to fool around with the message anymore, but move to the next message (or out of the area, or whatever) immediately, or the 'received' bit will be set again before you know it :)

ALT-D, D, : Delete message

Delete the current message. TimEd will first ask a confirmation.

ALT-W, W : Write to file

Write the current message to a file. TimEd will ask for a filename to write to. You can also put (for example) LPT1, PRN as the 'filename' to send output to the printer.

ALT-P, P : Print message (hardcopy).

This will send the current message to your printer, as defined by the 'Printer' keyword in timEd.cfg. If no 'Printer' statement was found in timEd.cfg, it will default to LPT1.

ALT-M, M : Move, copy, forward

This will let you move, copy or forward the current message to another area.

It will first show you the Area Selection Screen, where you can pick the destination area of your choice.

The 'forward' option will let you edit the header (TO: and Subj: fields) before dumping you in the editor, where you can edit or add to the body of the message, before saving it.

Differences between copy/move and forward:

Copy/Move.

These options will change VERY LITTLE of the message. SEEN-BY's, PATHs and VIA lines will be stripped. In addition 'sent' and 'scanned' bits will be toggled off. Nothing else is changed: the date/time is still the same, the MSGID is still the same etc. So this does indeed what the function says: copy (which is not change!).

This will make a message ready to be exported (again?) by your tosser, but note that if you are really sending it out AGAIN, dupe checkers may catch it as a dupe (as they should, of course, as it is a dupe).

If you copy/move a batch of messages (i.e.: copy/move tagged messages), even the above is not done and the message remains 100% unchanged. The reasoning behind this being, that a mass move/copy will be to an archive area and never intended to be exported again.

Also note, that echomail messages do NOT have a origination and destination address in the header. If you move/copy an echomail message to netmail (that *is* supposed to have useful address fields!), the address fields will show random values!

Forward.

Forward generates an entirely new message, with an entirely new header, with a correct name and address, origin, a new MSGID etc. In

all respects this is a new message, it just shows the body of an old message. In nearly all cases (except copy/move to an archive area and some exceptions) this is the function to use!

ALT-V, V, ALT-K, K : Toggle kludges

This will let you toggle the display of kludges (for this purpose, that also includes SEEN-BY lines). The default (at startup) can be set in timEd.cfg, using the 'showkludges' keyword.

ALT-J, J : Jump to DOS.

This will let you 'shell' to DOS. Type 'exit' to return to timEd.

ALT-I, I : Message & Area info

This will show a pop-up box with some info about the current message and the current area.

This will also show the 'setting' for a Squish style area: the maximum number of messages to keep in this area, the number of messages to skip when purging and the number of days to keep messages. From within timEd, you can set these parameters by pressing 'C' and editing the values to whatever you like.

'+' : Go to next area with new mail.

Pressing '+' will move you to the next area that has new mail. Using this you can quickly go from area to area, without going through the 'Area selection screen'.

If timEd cannot find a 'next area with new mail', it will drop you on the 'Area selection screen'.

'-' : Go to previous area with new mail.

This function is similar to '+', but moves to the previous area with new mail.

CTRL-<ENTER> : Place bookmark

This will place a mark on the current message. TimEd saves the position you are currently at, and when you later press <TAB>, timEd will go back to the message that you marked.

With this function, you can mark a message, then move around within the message area as much as you like, and go back to the message you started with, by simply pressing <TAB>.

Whenever you enter an area, a bookmark will automatically be placed on the current (i.e.: the lastread) message.

<TAB> : Go back to 'bookmarked' message.

This key will let you go back to the message you earlier marked with CTRL-<ENTER>.

<space>

By pressing the spacebar, you tag (mark) the current message. This will be indicated by the text '[marked]' at the right of the screen (in the header).

You can tag as many messages as you like in an area (up to 16000), in order to later perform a certain action on these tagged messages (like copying, moving, deleting, writing to file etc).

When leaving an area that still has tagged messages, timEd will ask you whether you want to remove these tags or preserve them. That way you can even leave an area, and come back later and still find your 'tags' there. It does take some memory though (this might be a concern for people in tight memory situations).

Moving to the next/previous marked message can be done by pressing ctrl-up and ctrl-down.

F1 : Help

This shows a short help-screen.

F9 : Lookup FROM: NAME of message

This will let timEd do a nodelist lookup for the name that is found in the from: field of the current message. (Note that this also works in echomail areas!).

If you're reading a message, and would like to know more about the person that wrote the current message, you can force timEd to look in the nodelist.

If timEd finds the name and pops up a window with the name(s), you can press <tab> to get even more details about the node.

F10 : Lookup origin ADDRESS of message

This will let timEd do a nodelist lookup for the origination address of the current message. (Note that this also works in echomail areas!).

If you're reading a message, and would like to know more about the person that wrote the current message, you can force timEd to look in the nodelist.

If timEd finds the address and pops up a window with the node, you can press <tab> to get even more details about the node.

Shift-F10 : Lookup TO: NAME of message

This function is similar to the two functions above, but will look up the name in the to: field.

ALT-X : Exit timEd

This will return you to the DOS prompt. If defined in timEd.cfg, an echotoss.log file will be generated at this time.

CTRL-N : Change current Username.

Pick one of the usernames defined (by the NAME keywords) in timEd.cfg. The name you pick will be used to enter new messages. It will be the default to use until you leave the message area.

CTRL-A : Change current Address.

A function similar to CTRL-N, but to pick one of the defined ADDRESSES.

ALT-H, H : Edit 'Hello' strings

This will show a pop-up box, where you can edit the 'hellostring', the 'rephello' string, your 'signoff' and 'origin line' FOR THE CURRENT AREA ONLY.

With this you can customize your setup on a per-area basis!

{+} The edited strings can be saved in the message area, so timEd can continue using them at a later time as well. Whenever timEd detects a 'custom setup' file *in a message area*, it will use the strings contained in that file (*only* in *that* message area), instead of the defaults defined in timEd.cfg.

TimEd saves the info in:

<areaname>.SQT for Squish areas,
timed.dat for *.MSG areas, and
<areaname>.JTI for JAM areas.
MSG##.TIM for Hudson areas (## is the board number).

The origin is saved in:

<areaname>.SQO for Squish areas,
origin for *.MSG areas, and
<areaname>.JTO for JAM areas.
MSG##.TIM for Hudson areas (## is the board number).

ALT-U, U: Utils (Message base maintenance).

There's one option that is present for all types of areas: Execute mail processor.

This command will write out the echotoss.log (if applicable) and execute 'mail.bat' (or mail.cmd for OS/2 systems). You can use this to spawn your echomail processor, without the need to exit timEd.

For *.MSG areas, the only other option is to renumber. TimEd will start a batchfile called renum.bat, with one parameter: the message dir for this area (like: renum.bat c:\binkley\netmail). In this batchfile you can call your favorite renumber program (maybe Bob Hartman's renum, I think the batch should then be "renum -R %1"), and maybe run other programs (like scanbld.exe for Max sysops) or touch semaphores or whatever.

For Squish areas, there are five other choices: Pack, Re-Index, Fix, Inspect and UNdelete. This will call SQpack, SQreidx, SQfix and SQinfo respectively (with the message area as parameter), or undelete all deleted messages in an area.

See to it that the programs (SQpack etc) are in your path. The programs are *not* called with the extension .EXE, so you *could* place a file called (for example) sqpack.bat (or .CMD for OS/2) in your path instead, and call SQpack/386/DPMI instead :)

For those who like to fiddle: SQpack is actually called with two parameters:

```
SQPack <dir/basename of area> <areatag>
```

SQpack.exe will not use the second parameter, but if you make a batchfile called SQPack.bat (dos) or SQPack.cmd (OS/2) you can use the second parameter, for example to link the message area as well, like this:

```
sqpack.bat:
sqpack.exe %1          <-- Don't forget .EXE!
squish link -f%2
```

For JAM areas, there is an 'Inspect' option. This will call a program called JAMINFO, with the path + basename of the current area as the only command line parameter. JAMINFO is a diagnostic utility that will scan JAM areas for errors.

Some considerations:

If you have a Squish area stuffed away in a directory that has a dot in it (like c:\msgs\sysop.028\squfile), SQpack will get confused (only SQpack, not the other utils) and think you specified an area.dat file on the commandline. This will happen if you would do this:

SQpack c:\msgs\sysop.028\sqdfile

.. which is what timEd does :) So it won't work for such areas.

CTRL-F : File request generation.

When you press CTRL-F, timEd will try to find filenames in a message (usually seen in 'file announcement' messages for SDS etc).

Unlike other message editors, timEd does not (even try to) support specific 'announcement programs' by recognizing their 'formats'. There are many programs out there, and modern ones even allow you to design your own format. I think it's a waste of (programming) effort and code (size/speed) to try and support them.

It just looks for strings that look like filenames. This search is 'extension driven': timEd simply looks for file extensions :)
Currently recognized:

"arj", "zip", "arc", "pak", "lzh", "sqz", "com", "exe", "lha", "zoo",
"txt", "sdn", "sda", "ans", "jpg", "gif", "tar", "rar"

When you press ctrl-f, timEd will try to find those extensions in the message, and a list with all filenames will pop up on the right hand side of your screen. You can scroll through the list, select files with <enter> or <space>, select all with '+', unselect all with '-', accept the current selection with <ctrl-enter>. Just as with a file attach.

Limited message scrolling is still available: press <ctrl-pgup> and <ctrl-pgdn> to move the message (in case you don't even recognize the filenames, and the descriptions are somewhere else in the message).

In case timEd missed a filename, or if you simply want to add one not listed in the message (like FILES), press <INS> and type the name of the file. It will be added to the list, already selected.

If timEd can't find any files, it'll show a list with only "FILES" on it. You can also abuse this to just request FILES from someone that seems to have an interesting BBS (but is not announcing files in that message).

When you press <ctrl-enter> to accept, timEd will take you to the area selection screen, where you can pick an appropriate (netmail) area. It'll then let you edit the message header (maybe add crash, or maybe do a lookup on the SysOp name to pick another high speed line of the same system), and after you press <enter> it'll write the file request message(s) for you (empty, we don't want to write senseless "automatic file request" messages).

Watch your phone bill! :-)

6.3 The List Mode.

ALT-L, L : List headers, normal format.

ALT-B, B : List headers, no TO: field, broad subject field.

This will show you a list of message headers in the current area.

Use <cursor up>, <cursor down>, <page up>, <page down>, <home> and <end> to move around through the list.

Press <ENTER> or <cursor right> to read the highlighted message.

You can also press <TAB> to get a display of the full header of the currently highlighted message, without leaving the list mode.

<ESC> will take you back to the current message before you pressed ALT-L.

Other keys that are active in 'list mode':

<Space> : mark highlighted message

+ : mark range of msgs

- : unmark range of msgs

This will show a pop-up box, where you can enter the range of messages to be marked.

ALT+ : mark all messages in this area

ALT- : unmark all messages in this area

ALT-D, D, : kill all marked msgs.

ALT-M, M : move all marked msgs to another area.

ALT-C, C : copy marked msgs to another area.

ALT-W, W : write marked msgs to file.

ALT-P, P : print marked messages.

F1 : Show a short help screen.

ALT-J, J : Jump to DOS.

ALT-S, S : Switch between 'broad' and normal subject width (for Usenet messages).

6.4 TimEd's Find feature.

ALT-F, F : Find message

This will show the Find screen, where you can set the criteria for the search. It looks like this:

```

timEd - Search configuration manager
-----
timed   (B--SK ---) & netmgr   (B--S- ---)
wimm    (B--S- ---)

-----[ Search options: ]-----
Areas : All      Messages: New      Direction: Forward
Action: Read

```

There are two main parts in the screen:

- A. The biggest part is the part where the strings that timEd will search for are shown. It consists of the strings themselves, followed by the search options for that item between brackets [an example: "timed (B--SK ---)"]. More than one item can be combined.
- B. The smaller part of the screen, at the bottom. This is where the search options that do not belong to one specific search item are shown. This includes things like which areas to search, and what the direction of the search is.

Part A, the search strings.

AND and OR searches.

TimEd is capable of doing 'AND' searches (that is: search for more than one string, and all strings must be present for the message to be consider a 'match' for the search).

It can also do 'OR' searches (search for more than one string, if any of these is present, the message is considered a match).

AND and OR can also be combined in a search (string A and string B must be present, or string C must be present in a message).

TimEd displays the elements of an 'AND' search on 1 line, like this:

```
timed   (B--SK ---) & netmgr   (B--S- ---)
```

This line holds two elements, separated by the AND token ('&'). Both elements consist of a string to search for ('timed' and 'netmgr' respectively), and the search options for that item (between brackets, let's ignore the search options for now, we'll get to them later).

The example above means: a message must both contain 'timed' AND 'netmgr' to be considered a 'match'.

A line can also contain just one element, in that case only that element has to be present for a match.

The elements for an 'OR' search are shown as different lines, like this:

```
timed (B--SK ---)
wimm (B--S- ---)
```

The above means: a message must contain 'timed', OR 'wimm' to be considered a match.

Finally, AND and OR can be combined, where we get the screen that we saw earlier:

```
timed (B--SK ---) & netmgr (B--S- ---)
wimm (B--S- ---)
```

This means: a message must contain 'timed' AND 'netmgr', OR it must contain 'wimm', in order to be considered a match.

Search options.

The search options are shown behind the string, for example:

```
timed (B--SK ---)
```

Here the search options are B, S and K.
The following options are possible:

- B - Search for the string in the body of the message.
- F - Search for the string in the from: field of the message.
- T - Search for the string in the to: field of the message.
- S - Search for the string in the subject field of the message.
- K - Search for the string in the kludges of the message.
- W - Search for 'whole words' only (not a part of a word or substring).
- C - Search is case sensitive.
- ! - String must NOT be present.

Certain letters, that represent certain options, will always show up in the same location. If the 'search body' option is active, the 'B' that indicates this will always be the first letter. If you do not want to search the body, the first 'letter' of the search options field will be a minus sign ('-').

Entering search strings and options.

This is very easy. Just press <INS> to enter a new string to search

for, or press <enter> on a certain line that you wish to change. Press to delete an existing item.

For editing, you will be presented with the following pop up menu:

```

timEd - Search configuration manager
-----
timed      (B--SK ---) & netmgr   (B--S- ---)
wimm      (B--S-
-----
String:           Options:
-----
timed             B--SK ---
netmgr            B--S- ---
-----
-----[ Search options: ]-----
Areas : All      Messages: New      Direction: Forward
Action: Read

```

In this popup window, you can enter the separate elements of the 'AND' search and it's options. You can enter up to 10 elements for an 'AND' search.

Once again, you can add neww items with <INS> and edit existing ones with <enter>. You can delete an existing item with . Once ready with editing, press <ESC> to leave this part.

First, you can type in the string to search for. The press <enter>, and you will get the following pop-up menu to set the search options:

```

String:           Options:
-----
test             Search body           -
                 Search from: field
                 Search to: field
                 Search subject
                 Search kludges
-----
                 Whole words only
                 Case sensitive search
                 Should NOT be present
-----

```

You can scroll through this menu, and press <enter> to toggle a certain option 'on' or 'off'. Once ready, press <esc> to leave this

part.

Part B. Setting the search options.

From the main screen, you can press ALT-O to set the search options. You will be presented with the following menu:

Areas	All
Messages	New
Direction	Forward
Action	Read

You can scroll through the options, pressing <enter> to switch between the different possibilities.

To select an 'Action' you will once again get a pop-up menu, where you can choose what you want to do with messages that are found:

Read	- Read the found message(s).
Mark	- Mark ('tag') the message(s).
Delete	- Delete the message(s).
Write to file	- Write all found messages to a file.
Copy	- Copy messages to another area.
Move	- Move the messages to another area.

{+} Since it can be a lot of work to enter a bunch of strings with their options, timEd offers you the possibility to save the search profile of a certain area.

If you press ALT-S, timEd will permanently save the search profile. Whenever you enter the Find function in that particular area again, timEd will read in these options from disk. This way, you can regularly do quite complex searches in certain areas without any trouble.

Once all options are set, and strings to look for are entered, you can start a search by pressing ctrl-<enter>.

TimEd will then search where you told him to search, for what you told him to search for. Whenever it finds a match, it will do with it what you told him to do with it :-)

It will search headers, kludges or bodies, all areas, the current area etc. And mark, copy, move, delete, write to file or let you read every match it finds.

During the search, you can influence timEd's behaviour:

<cursor left>: timEd will search backward, even if it was currently scanning forward.
<cursor right>: timEd will search forward, even if it was currently scanning backward.
'+' Skip rest of this area, and go on scanning in the next area (searching forward from now on).
'-' Skip rest of this area, and go on scanning in the previous area (searching backward from now on).
<esc> Abort the scanning.

These keys can be using during the actual scanning process. They can also be used when you are reading a certain message that timEd found (evidently only if you set the 'action' to 'read' and not something else like 'tag').

7. TimEd keyboard reference

This section simply lists all available key combinations that should do something useful in timEd. Features marked with {+} are for registered users only.

In general: F1 will give you some help in several places.

Area selection screen:

```
ALT-S      : Scan unscanned areas for new mail
ALT-T      : Scan all areas (even if already scanned) for new mail
ALT-P {+}  : Personal mailscan in all areas
            When a message is found:
            ALT-R : Reply
            ALT-N : Reply in other area
            etc.
<ESC>     : Interrupt scan
ALT-J     : Jump to OS (Shell)
ALT-X     : Exit timEd
<ENTER>   : Enter highlighted area
<right arrow> : Enter highlighted area

ALT-L     : Enter area in L)ist mode
<space>   : Tag area.
ALT-W     : Write out current 'tagset'.
ALT-W     : Read in 'tagset' from disk.
ALT-M     : Change current 'view' on areas (all, new, tagged etc.)
F1        : Help
```

Message reading screen.

```
@ = hold down ALT key.
^ = hold down CTRL key.
Sh = hold down Shift key.
```

```
home      : Top of message
end       : Bottom of msg
@E, E, ins : Enter message
@R, R     : Reply to message
@T, T     : Reply, fast (don't edit header)
@N, N     : Reply in another area
@O, O     : Reply 'followup' (reply addressed to the same
            person).
@C, C     : Change message
^H        : Change header
@S, S     : Change attributes
@D, D, del : Delete message
```

```

@P, P      : Print msg
@W, W      : Write to file
^U         : Unreceive msg (toggle 'received' bit off).
^B         : Bounce reply (reply, don't quote but show original
           : message)
@M, M      : Move/Copy/Forward
space      : Mark message
@L, L      : List headers
@B, B      : Broad list
@H, H      : Edit 'Hello' info
@I, I      : Msg & Area info
@F, F      : Find message
^A, ^N     : Change address or change name
^F         : Generate file request from files found in this
           : message.
@V, K      : Toggle display of kludges
@X         : Exit timEd
@U, U      : Area maintenance menu
@J, J      : Jump to OS (Shell)
'+', '-'   : Goto Next/previous area with new mail
^up/^down  : Goto Next/previous marked message
<number>   : Goto msg # (specific number)
right, left : Next, previous msg
enter      : next message
^end, ^home : Last, first message
^right, ^left : goto reply, original
@+/@right  : next reply (JAM)
esc, @A    : Quit area
^enter     : Set bookmark
TAB        : Return to Bookmark
F10        : Lookup origination address of message in nodelist
Sh-F10     : Lookup TO: name in nodelist
F9         : Lookup FROM: name in nodelist

```

Message list screen.

```

<ENTER>, <right> : Goto highlighted msg
up, down         : Move up, down in list
page up, down    : Move page up, down in list
home, end        : First, last msg in area
<SPACE>         : Toggle mark of highlighted message
<TAB>           : Show full header of highlighted message
+               : Mark range of messages
ALT+            : Mark all messages
-               : Unmark range of messages
ALT-            : Unmark all messages
ALT-D, D, <del> : Delete marked messages
ALT-M, M        : Move marked messages
ALT-C, C        : Copy marked message
ALT-W, W        : Write marked msgs to file
ALT-S, S        : Switch between broad/normal list
ALT-J, J        : Jump to OS (Shell)

```

<ESC> : Return to message reading screen
F1 : Help

The 'find feature' screen.

ALT-R : clear all search strings
ALT-S {+} : save current search arguments
ALT-O : edit generic search options
ctrl <ENTER> : start search
When a match is found:
ALT-R : Reply to msg
ALT-N : Reply other area
etc.
Right arrow : search for next match

While searching or match found:

<ESC> : Interrupt search
+ : Skip rest of this area, continue search in next area
- : Skip rest of this area, continue search in previous area
<right> : Go on searching forward
<left> : Go on searching backward

File attach selection.

<ENTER>, <SPACE> : Toggle mark on file
+ : Mark all files
- : Unmark all files
ctrl <ENTER> : Accept selection
<ESC> : Go back to subject editing

The Internal editor.

ctrl-<cursor right>	: jump word right	ALT-S	: Save Message
ctrl-<cursor left>	: jump word left	ESC	: Abort message
ctrl-<home>	: Top of message	ALT-J	: Jump to OS (Shell)
ctrl-<end>	: End of message	ALT-I	: Import file
ctrl-<page up>	: Top of screen	ALT-H	: Toggle HCR display
ctrl-<page down>	: Bottom of screen	ALT-Z	: delete to end
ctrl-	: Delete to end of line	ALT-L	: Mark block border
ctrl-T	: Delete word right	ALT-U	: Unmark block
ctrl-<backspace>	: Delete word left	ALT-M	: Move block
ctrl-Y	: Erase line	ALT-C	: Copy block
ctrl-U	: Unerase line	ALT-D	: Delete block
F1	: Help	F4	: Dupline.

8. Effective use of timEd: some features explained

8.1 Creating netmail messages.

This is a special section on the netmail area. Netmail is different from echomail in many ways, and timEd behaves differently when you enter netmail.

While in the netmail area, you will also have to enter a fidonet address, in addition to the name of the recipient.

TimEd can help you in several ways, especially if a Version 7 nodelist, a FrontDoor style nodelist or a fidouser.lst format list is present.

Changing your name and address.

First of all, when timEd lets you edit a header, it will put you in the To: name field. The From: name (your name) and address (your address) will already be entered, using the defaults that are active for that area.

If you used CTRL-A (change address) or CTRL-N (change name) just before entering a message, timEd will use that name/address.

While editing a header, however, you can still edit that info. Just press <cursor up> and you will go to the origination address and name fields. There you can edit them to be whatever you want, or press TAB to pick one of the names/addresses you entered in timEd.cfg.

Entering a (TO:) name:

When you enter a name (or part of the last name) in the TO: field and press <ENTER>, timEd will try to find that name in the nodelist and when found it will show a pop-up box with name(s) and address(es) it found.

TimEd will try to find all matches to the string you typed. So typing 'essen' might give you both 'Gerard van Essen' and 'Martin Essenburg'. If more than one match is found, timEd will show a list with matches and let you scroll through the list.

If you want to know a bit more about a certain node, just press <tab> and timEd will show you a few extra details about that particular node.

TimEd will stop searching when more than 50 matches are found.

If you then press <ENTER>, timEd will fill in the selected address for you. If you want none of the addresses that are shown, press <ESC>, timEd will then let you fill in an address manually.

This function works for all supported formats: Version 7, FrontDoor nodelist and fidouser.lst.

TimEd will first look for a fidouser.lst style list. If such a list is not present, or if no match is found in this list, it will try the FD nodelist. If still no match is found, or if there is no FD nodelist at all, it will try the Version 7 nodelist.

The fidouser.lst is searched first. Since you can have more than one nodelist format present, you could use this to make a short list of people you regularly send mail to. Looking through this short list can be done very fast, of course. If nothing is found, timEd will go to one of the other nodelists (if present).

This is only interesting if your system is very slow, of course. Even on a 10 Mhz 286 (28ms HD) both V7 and FD nodelists are **very** fast. But your mileage might differ.

The above is valid when entering **new** netmail. When replying to a message, or when changing a message, timEd will see that an address is already present, and it will normally only check if that node (the ADDRESS) exists. If it doesn't, timEd will display "Unknown Address".

Additionally, a 'fresh' name lookup will be done if you change the TO: name field, or if you press F2 when you're in the TO: name field.

Entering an address:

In the TO: field, you can also enter an address (!). Again, timEd will try to locate that address in the nodelist and try to fill the name in for you.

There are several shortcuts when entering an address. When you don't type in a 3D/4D address, timEd will try to fill in the missing parts with the default from **your** address.

An example:

My address is 2:281/527.

If I now enter: 281/1, so with a missing ZONE number, timEd will fill in **my** zone number (that's 2, so the address becomes 2:281/1).

In this case, I could even have entered: 1, because then timEd would have filled in net 281 (my NET) and zone 2 (my ZONE), producing the wanted address (2:281/1).

I can also enter .1, (so missing zone, net and node info) and timEd will fill in my zone (2), my net (281) and my node (527), producing

2:281/527.1.

After 'expanding' the address, timEd will try to find it in the nodelist.

This function only supports the Version 7 and FrontDoor nodelists. For the fidouser.lst lists, this is not possible.

Generating a file-request message.

When you enter a message with the 'file request' attribute, timEd will ask if you want to include any text in the message. If you send a file request message with a body that is empty, it will usually be automatically killed on the destination system. This way the SysOp of the destination system won't have to wade through all these useless messages that consists of one line only ('This is an automatic file request').

Generating a file-attach message.

TimEd will try to detect whether you entered a path specification in the 'subject' field, and set the file-attach bit automatically if you did. Of course you can also set the file attach bit manually.

TimEd will then check whether the filename(s) entered on the subject line exist, and if you entered a wildcard and more than one match is found, timEd will show a list with matching files. You can then 'tag' the files you want to attach. If you tag more files than fit on the subject line, timEd will generate extra messages.

A file-attach list may look like this:

--

```
Date : 25-
From : Ger Current filespec: c:\files\c\t*.*
To   : ROn
Subj : c:\files\c\t*.*
```

TABX.C	3 K
TADV2DEM.ARJ	53 K
TC1TUTOR.ARJ	105 K
TC2TUTOR.ARJ	83 K
TDT-2.ZIP	268 K
TDT-3.ZIP	249 K

--

You can now 'tag' files with <ENTER> or <SPACE>, move around with <cursor up> and <cursor down> etc, and accept the currently tagged files with <CTRL-ENTER>.

You can select all files in one action by pressing '+', and un-select them all by pressing '-'.

This process will be repeated for all entered file specifications.

8.2 AKA matching.

This option is interesting if you have more than 1 address. If you set the 'AKAmatching' keyword in timEd.cfg to 'Yes', timEd will try to find the most appropriate address to use when writing a message.

Say, for example, that you have two addresses: 2:281/527 and 60:100/112.

If you write a messages to 2:500/133, you probably want to use your 2:281/527 address.

If you write a message to 60:100/1, you probably want to use your 60:100/112 address.

In this case, timEd would try to find the address (AKA) that 'matches' the destination address best.

It first looks for a matching zone, and if more than one match is found, it'll try to find an address where both 'zone' and 'net' match. If there is still more than one match after that, it will just take the first match.

This is always active (in the netmail area): when writing a new message, a reply, or changing a message.

If timEd selects an address different from the address that was currently used (so if anything changed), it will say so ("AKA matched") just below the addresses on the screen.

If you want to force it at a certain moment, press F2 when on the TO: address field.

TimEd will stop matching the AKA's, as soon as you changed your address manually. So if you don't agree with timEd's AKA matching in a certain situation, just manually change it (while editing the header, you can 'cursor up' to the from: address field) to what you want it to be, and timEd will leave it at that.

It will not match AKA's anymore after you did that. (Of course, when you enter another new message after that, the game will start again :)

AKA matching is only active when you enter a new message. When changing a message (or header), AKA matching won't kick in. You can, however, still force AKA matching by pressing F2.

8.3 TimEd's Carbon Copy (CC:) feature.

TimEd can also make 'carbon copies' of netmail messages you write. It can happen that you want to send an exact copy of a message to someone other than the original recipient of the message.

For example, when I release a new beta version, I should have to write the same netmail message for every beta tester.

Now say I want to write the same message to Kasper Kwant, ROn van der Nagel and Frank Troost.

I would enter the message, addressing it to Kasper Kwant, and on the first line of the message I would write:

```
cc: ROn van.der.Nagel, Frank Troost
```

When the message is saved, a copy would be generated for both ROn and Frank.

Rules for the cc: line(s):

- * It must start on the first line of the message.
- * If all names do not fit on one line, proceed on the next line, and put cc: in front of it again.
- * You can use a full name (it will be looked up in the nodelist), a 'macro' (it will be expanded) or an address (it will be looked up in the nodelist). This is exactly the same as you can enter at the TO: prompt of a 'normal' netmail message.

Nodelist lookup is a bit different, though: if timEd only finds one match for a certain name, it will use that to generate the message. If more than one match is found, it will prompt you with a list.

In addition, if you type a full name which is **not** present in the nodelist (or if you don't want timEd to look in the nodelist, for whatever reason, like speed), you can also specify **both** name and address, separated by a '#', like this:

```
Gerard van Essen#2:281/527
```

And finally, if you regularly send CC:'s to the same bunch of people, you can put their names, macro's or addresses in a file, one per line, and specify that filename in the cc: line, like this:

```
<c:\timed\names.cc
```

The '<' is chosen as it looks like 'redirection' in DOS :-)

The contents of such a file could be:

--=
Kasper Kwant
ROn van.der.Nagel
Frank Troost
pietje#2:281/527.29
--=

Of course, you can combine all this. A full example (let's say that I have a macro for my HUB, called 'hub'):

cc: 1, hub, Kasper Kwant, <names.cc
cc: ROn van.der.Nagel, .23

- '1' will be expanded to myzone:mynet/1 (in this case 2:281/1)
- 'Kasper Kwant' will be looked up in the nodelist.
- The file 'names.cc' will be read and all persons listed in that file will get a copy.
- ROn van.der.Nagel will be looked up in the nodelist.
- 2:281/527.23 will get a copy too.

All CC: messages that are generated by timEd will be marked "kill/sent", so you won't have lots of copies of the same message in your netmail area. The attributes of the original message will be left untouched, of course.

Before writing out these messages, timEd will expand the CC: line(s) you entered into <name> (<address>) form, like: Gerard van Essen (2:281/527), so that your macro's and addresses are converted into something everybody can understand.

You can prevent this from happening, by using the 'CCVERBOSE No' in timEd.cfg. In that case, timEd will only show the number of carbon copies sent, not the names addresses of the other recipients.

Please don't abuse this feature by sending junk mail to everyone..

8.4 TimEd's crossposting feature.

TimEd can crosspost messages. Crossposting is a function that will place a copy of a message that you write in several other message areas.

This is a heavily abused function (we didn't split up FidoNet into different areas for nothing!), but it can be useful in some cases.

You can make timEd place copies into other areas by writing...

XC: <areaname>[, <areaname>, <areaname>]

.. at the top of the message. A copy of your message will be placed in all areas mentioned behind the XC:, where <areaname> should be a valid areatag.

Example:

```
From: Gerard van Essen
To : All
Subj: New timEd release!
```

```
-----
XC: artware, enet.soft, software.028
```

Hello All!

[....]

This will cause copies of this message to be placed in 3 other areas:
artware, enet.soft, software.028.

8.5 Message attributes

After editing a message header, timEd surprises you with the following screen:

```
[P]rivate          [O]rphan
[C]rash           [Q] ret. receipt re[Q]uest
[A]ttach file     [Y] return receipt
[R]equest file
[K]ill/sent       Extra 'flags':
[L]ocal
[H]old            [E]rase file when sent
sca[N]ned         [X] archive file when sent
[F]orward         [I]mmediate
[S]ent            [T]runcate file when sent
recei[V]ed        [W] Lock
[D]irect          [M] confirm receipt request
[U]pdate req.
```

This screen lists several message attributes that can be used to influence the handling of your message by echomail processors, mailers etc.

For echomail areas, only the following attributes are usually useful and needed:

- Local - This tells other software that this message was generated locally (i.e.: on this system). In order to let the echomail processor send a message out, the Local bit is often required. It does **not** mean that the message shouldn't be sent out (so: stay 'local' on that system), on the contrary. TimEd automatically sets this bit on messages it creates.
- Sent - This indicates the message is already (actually) sent, or that the message was already processed by the echomail

processor (that is: 'scanned out' and placed into a mailpacket).

Scanned - This is practically the same: the message was already scanned (processed) by the echomail processor. Some msgbase format use a 'Sent' bit, others (Squish) use 'scanned' to indicate more or less the same.

Received - This indicates that the message was received by the person it was addressed to. So, if you write a message to 'Pietje Puk', the 'received' bit will be set after Pietje Puk actually read your message.

In netmail messages, many other attributes can be used, in addition to the four attributes mentioned above. These attributes are not always available, but depends on support by other software (like your echomail processor and/or mailer) and message base format (some formats simply do not support a certain attribute).

Additionally, some programs implement the attributes somewhat differently from other programs. Especially your mailer manual could reveal interesting details in this respect.

Private - This indicates that it is a Private message, that should only be read by the person who wrote the message, and the person to whom the message is addressed.
BBS packages usually honour this attribute and won't show the message to anyone else (except the SysOp of the system, who can always see everything :-)

Crash - This indicates that this message is 'high priority'. It should be sent out quickly, often regardless of schedules or cost. The message will definitely not be routed through another system.

Attach - File attach. This indicates that a file is attached to this message. You can use this to send files to the person the message is also sent to. When the message is delivered on the recipient's system, the file is sent as well. It 'travels with' the message. The name of the attached file should be placed in the 'Subject' field of the message header. File attach messages can usually not be routed, you must deliver it directly.

Request - File request. By sending a file request message to another system, you are requesting (asking for) a file. When your mailer (or echomail processor) 'sees' that you are sending out a file request message, it will take certain actions, and ask for the file to be delivered. The name of the file you want to receive should be

placed in the message header's 'Subject' field.
File requests can usually not be routed, you must directly call the system, that has the file that you want, to receive it.

- Kill/sent - This indicates that the message should be deleted from the message base, as soon as it is delivered on the destination system, or as soon as it is packed into an outbound mailbundle.
- Hold - Hold for pickup. Hold (don't send out) this message, until the 'hold' attribute is removed, or until the person that the message is addressed to calls you to pick it up.
- Forward - This attribute is placed on messages that were not created on your system, and are not destined for your system either (in other words: this is a 'routed' message, that someone else generated, and that is travelling through your system to it's destination).
- DIR - Direct. This message must be delivered directly at its destination. It should not be routed through other systems.
- Update request - This is similar to a file request, but the requested file will only be sent, if the file on the other system is newer than the copy of the file that you already have on your system.
- Orphan - Indicates that this message is destined for a system that does not appear in the nodelist (Frontdoor environment usually).
- RRQ - Return receipt request (RRQ). This requests a return message to be sent by the mail processing software of the system the message is destined for, to confirm arrival of the message on the system.
- Return receipt - A return receipt message that is sent in reply to an RRQ has this attribute.
- KFS - Kill File Sent, Delete the attached file as soon as the message (and file) is delivered at its destination.
- IMM - Immediate, high priority mail. Do not route, send out immediately, do not care about schedules or cost.
- TFS - Truncate File when sent. When the message (and the file) is sent, the file will be truncated (the length of the file will be set to 0 bytes).

- Lock - This message is locked. Do not pack, send out, delete or whatever.
- CFM - Confirm Receipt Request. This asks for the software of the destination system to send a confirmation message, as soon as the recipient has actually READ the message.

Not all attributes are supported by all programs and message base formats. Some attributes can be represented by a setting a bit in the attribute field in the message headers, others have to be 'emulated' using a FLAGS kludge. This makes the situation somewhat unclear. Below is a table that tries to help you a bit:

	JAM	Squish	Hudson	*.MSG
Local	x	x	x	x
Sent	x	x	x	x
Scanned	-	x	-	-
Received	x	x	x	x
Private	x	x	x	x
Crash	x	x	x	x
Attach	x	x	x	x
Request	x	x	F	x
Kill/sent	x	x	x	x
Hold	x	x	F	x
Forward	x	x	-	x
DIR	x	F (*)	F	x
Update request	-	x	-	x
Orphan	x	x	-	-
RRQ	x	x	x	x
Return receipt	-	x	x	x
KFS	x	F	F	F
IMM	x	F	F	F
TFS	x	F	F	F
Lock	x	F	F	F
CFM	x	x	F	F

Entries that are marked with 'x' are always available.

Entries marked with '-' are not available.

Entries with 'F' can only be represented with the FLAGS kludge, which means that you have to have 'ArcMailAttach Yes' in your configuration in order to make them available from within timEd. Otherwise, it will allow you to set them, but when the message is written to disk, timEd will not add a FLAGS kludge, causing them to be lost.

(*) The DIRECT attribute is a strange thing. Squishmail (the tosser, that is) can 'simulate' that attribute by setting BOTH the 'crash' and 'hold' bits. Squishmail will also NOT recognize the FLAGS kludge.

On the other hand, some other programs DO support the FLAGS kludge, and will certainly not know what to do when both 'crash' and 'hold' attributes are set.

In environments where Squish takes care of the netmail (Squish combined with Binkley), you need to set ArcMailAttach to 'No' and timEd will use the 'simulated' DIRECT bit.

If you set 'ArcMailAttach' to Yes, timEd will write out the FLAGS kludge (all other cases, even Squishmail if combined with FrontDoor).

Confusing? Yeah. But I didn't invent it! :-)

8.6 Using the internal editor

If you put 'Editor Internal' in timEd.cfg, you will be able to use timEd's internal editor. This section describes the main commands you can use within the editor.

First some general information:

- * The editor automatically wraps lines. You don't have to press <ENTER> at the end of a line, in order to go to the next line. Just keep on typing, timEd will place you at the beginning of the next line automatically, together with the word you were typing (if it doesn't fit on the previous line).
- * The editor also actively rewraps paragraphs if you insert text in the middle, or delete text somewhere. TimEd will rewrap immediately, so you see the result of your actions at once.
- * The editor is 'quotesmart': it know the difference between a line that is a quote, and a normal line. It does not only show quoted lines in a different colour, but also handles them differently.

For example: when you press <ENTER> in the middle of a quoted line, timEd will move the rest of the line down, but will also place a quotestring in front of it!

Similarly, if you delete text in a quoted line, and press at the end of that line, timEd will try to append the next line to the current line. It will, however, not append the entire line, but will first strip the quotesting at the beginning!

All this may sound a bit difficult, but it is meant to make editing quotes just as easy as editing quoted lines. It may be difficult to explain, but when you use it, you will see that it feels 'natural'! Try it!

First of all: help is always available by pressing the F1 key!

The editor is mainly a standard editor: you can just type the text you want to enter, use the cursor keys (and page up / page down) to move around, and the <INSERT> key to switch between the 'Insert' and 'TypeOver' modes.

There are also some other 'movement' commands:

```
ctrl-<cursor left>  :   jump one word left
ctrl-<cursor right> :   jump one word right

ctrl-<home>          :   Go to the top of the entire message
ctrl-<end>           :   Go to the end of the entire message
ctrl-<page up>       :   Go to the top of the screen
ctrl-<page down>    :   Go to the bottom of the screen
```

Some commands to delete text:

```
ctrl-T              :   Delete the word to the right of the cursor.
ctrl-<backspace>    :   Delete the word to the left of the cursor.

ctrl-Y              :   Erase the entire line.
ctrl-U              :   Unerase a line that was erased with CTRL-Y.
ctrl-<del>           :   Delete from the cursor to the end of the line.
```

Some block commands:

ALT-L : Mark block border.

Use this at the beginning and the end of a block. You can only mark entire lines, not a part of a line. The block functions will also work if you only mark one line.

ALT-U : Unmark block.

This removes the mark you defined earlier with ALT-L.

ALT-M : Move block.

Move a block to another position. The block is inserted below the line the cursor is currently on.

ALT-C : Copy block

Copy a block to another position. The block is inserted below the line the cursor is currently on.

ALT-D : Delete block

This will delete all the marked lines. They can NOT be undeleted with CTRL-U!

ALT-Z : Zap quoted lines until end of text.

This will delete the quoted and empty lines below the cursor, until a

non-quoted, not empty line is found.
This is useful when editing a quoted message. With this, you can 'zap' all quotes until the end of the message, without deleting your signoff and origin lines.

Some general commands:

F4 : duplicate current line. An exact copy of the current line will be inserted on the line below the current line.

ALT-H : Toggle HCR display (HCR == Hard Carriage Return).

The editor has an option to display the 'hard returns' in the text.

In fidonet, 'hard returns' should only be at the end of a PARAGRAPH.

The internal editor will rewrap entire paragraphs, which means it will wrap until a line with a 'hard return' is encountered. Other fidonet systems (receivers of your message) will do the same.

The HCR display gives you total control over the location of any hard returns. If you don't want a certain part of the message to ever be rewrapped, you can place 'hard returns' at several places. TimEd can show you exactly where the HCR's are located!

ALT-J : Shell to OS

This will give you a shell to the operating system (DOS or OS/2).

ALT-I : Import file

This option will let you import a file from disk. A pop-up box will ask you for the name of the file to import.

You can enter a wildcard here (like c:\doc*.*). If more than one file is found, timEd will display a selection list where you can choose the file you want to import (by pressing <ENTER>).

If you enter a directory name (like c:\msgs), timEd will pop up a directory listing of that directory (in this case: c:\msgs*.*).

The directory lister sorts the files found in a directory, and allows you to move into other directories.

ALT-E : Edit header.

This will let you edit the header of the message you are currently working on. That way you can still change the attributes, or attach a file or something like that, even if you already started editing the body!

ALT-S : Save Message

This will end the editing session. The text you entered will be used to generate a message.

ESC : Abort message

This will end the editing session. The text you entered will be lost, and no message will be generated. It is possible to get a confirmation first, look at the 'Confirm_Edit_Exit' keyword for more info.

8.7 Using an external editor

Many people are very attached to a certain editor. They want to use that editor, and that editor only. TimEd will let a user do just that.

When the time has come to edit the message text, timEd will create an ASCII file on disk, and spawn an external program to edit that text. The user can now edit the message text. When ready, the text should be saved to disk.

After leaving the external editor, timEd will regain control, read the (edited) message text back in, and create a message from it.

This sounds to good to be true, and unfortunately it is not without problems..

Using an external editor has one big problem: most editors put a 'hard return' at the end of *every* line, not just on the lines where the user actually pressed <ENTER>.

In Fidonet, 'hard returns' should only be present at the end of a *paragraph*. The actual formatting (word wrapping) of the paragraph should be done by the message reader, according to the current screen size (usually, the screen has lines of 80 characters, but that is certainly not always the case! Think about older terminals, about 132 character mode, about proportional fonts [OS/2, Windows]).

Placing a 'hard return' at the end of every line (at position 80), would only use the first 80 characters of a line, even on a 132 character display! Unfortunately, there are quite a lot of programs that leave a hard return at the end of every line anyway. Especially QWK readers are notorious for that. It's quite a lot easier for the programmer though.

Many people even think that stripping 'hard returns' is wrong, because their text may not be formatted the way they wanted. Remember that not stripping them is even worse! How do you think your formatted text looks on a terminal with 80 or 132 columns?

So, timEd has to strip 'hard returns'. It tries it's very best, but sometimes fails miserably. But you can help timEd to keep your message 'tidy', read on!

One of the 'rules' is, that timEd will strip the 'hard return', if it is found at a position *after* <length of line> - 20 (so on a 80 character display, it would be 60). 'Hard returns' found after that position are supposed to be the result of the 'word wrapping' feature of an editor and are stripped.

There is an exception to this: if the *next* line is starting with a <space> or one of the following characters: *,., the 'hard return' will not be stripped. Usually there is some formatted text (in the case of '*' and '-' probably a list, in the case of a space it could also be a 'listing' of some code for a programming language), so that will be left untouched.

There is a way to prevent timEd from 'intelligently' stripping 'hard returns'. Usually it does what you want, but formatted text can be awfully messed up, when the individual lines exceed 60 characters in length (for example Binkley logfiles :-). TimEd will then strip 'hard returns' making a mess of your nicely formatted logfile or C-listing.

There are 2 methods to prevent timEd from stripping 'hard returns':

- Forcing a 'hard return' for one line:

Put a ~ at the end of a line. The ~ will be replaced with a 'hard return'.

- Forcing 'hard returns' for a sequence of lines:

Put ~~ at the beginning of a line (rest of line should be empty!) to start, and another one at the end. All of the text between those two lines will get a 'hard return' at the end of the line. The lines with the ~~ at the beginning will be stripped from the text.

Examples:

This line will certainly have a 'hard return' added to it.~

Now look at my nicely formatted logfile:

```
--  
~  
= 23 Sep 03:23:35 MAX DL-Z C:\Files\Tekst\Contrast.Lzh  
= 23 Sep 03:23:35 MAX Free DL: reimbursed 227 seconds  
= 23 Sep 03:23:35 MAX Free DL: reimbursed 51 Kb  
+ 23 Sep 03:23:59 MAX Jack Brons off-line. Calls=16, Len=6  
: 23 Sep 03:24:00 MAX End, v2.01 (5)  
~  
--
```

Keep this in mind, and experiment a bit. Just type some messages and check the results after timEd read it back in.

8.8 Reading messages 'gated' from Usenet/Internet

Some consideration was also given to people reading messages that originally came from Usenet/Internet 'newsgroups'.

FSC-0035

=====

TimEd supports the 'Replyto' and 'ReplyAddr' kludges as outlined in FSC-0035 (Fidonet Technical Specification - a proposal, actually). This is used for netmail replies. If you read gated Use/Internet messages that have this kludge (Fredgate can generate them), this will automatically address the netmail reply to your UUCP gate and put a TO: line at the top of your message.

'Broad' message L)ist display.

=====

For Use/Internet newsgroups the standard L)ist display is far from optimal: there, every message has 'All' in the TO: field (put there by the UUCP <-> Fido gate).

TimEd has two L)ist modes, a 'normal' one, and a 'broad subject line' one (you get it with Alt-B when reading a message). The latter has (as one would expect) a much broader Subject displayed, the TO: field is entirely zapped in that mode.

While in the L)ist mode, one can switch between the two styles with ALT-S (from S)witch..)

Usenetmacros

=====

TimEd has a special type of macro's available for message that need to be gated to Internet. See the 'Usenetmacro' keyword for further reference.

9. Message base formats explained

There are several different formats to store messages on a system. TimEd supports four of them: Squish, JAM, Hudson and *.MSG. What follows is an explanation of each of these formats. It is not my intention to give an in-depth discussion of the inner workings of these formats. That information may be very interesting for developers, but this document is not intended for programmers: it is meant for users, and as a consequence only the practical implications that are a result of the inner workings and design of the formats is discussed.

9.1 *.MSG

This is 'the original'. In the beginning stages, Fidonet was very small. The amount of messages that had to be processed was equally small. The format used to store the messages was very simple:

Every message area has its own directory. In this directory, the messages for that area are stored, each message in a separate file, called <number>.MSG, for example 55.MSG.

For an area that is called ARTWARE, the messages could be stored in C:\MSGs\ARTWARE, and message number 10 would then be C:\MSGs\ARTWARE\10.MSG.

This means that maintaining the message base can be done through standard file system operations: deleting a message is equal to deleting a file, copying a message is equal to copying a file, etc.

The lastread pointers are stored in a file called lastread (for the sysop) and usually lastread.bbs (for the other users, for those who are running a BBS).

The advantages and disadvantages of this format are closely related to the file system used:

Advantages:

- * This format is very simple. Implementing and supporting it is easy.
- * It is a very safe format: every message is in a separate file. If a file gets corrupt, this is only one message. Reconstruction of damaged message bases can be done through standard file recovery software (even DOS' undelete and checkdisk!).

Disadvantages:

- * In DOS, every file takes up at least the size of one cluster. On most harddisks this is 2 kilobytes. As a result, even the smallest message takes at least 2 Kb.

With several thousands of messages available on a system, this 'slack space' may add up to quite a large amount.

- * DOS gets very slow when there are a lot of files (>250) in one directory. On larger message areas, this will slow down even the most efficient programs.
- * Processing messages means constantly opening and closing a lot of files. This also slows things down.

OS/2's HPFS file system will help to relieve the two first disadvantages, but not enough to make it efficient.

9.2 Hudson

Hudson is more or less the exact opposite of *.MSG.

Where *.MSG stores all areas in a different directory, and every message in a different file, Hudson puts all messages from all areas in only one set of files.

This 'set' consists of the following files:

- * msginfo.bbs

This file is always 406 bytes long, and contains general information about the message base, like the total amount of messages, the highest message number and the amount of messages present in each different area (called 'board' in Hudson speak).

- * msghdr.bbs

This file contains the message headers (from: and to: fields, subject, date etc).

- * msgtxt.bbs

This holds the actual message text (the 'bodies' of the messages), including the kludges.

- * msgidx.bbs

This is an index that holds, for each message, the message number and the board number. This index can be easily searched for messages that are present in a certain board.

- * msgtoidx.bbs

This file holds a copy of the TO: field, for each message that is present in the Hudson base. In other words: for the entire message base.

In order to do a personal mailsan for a particular user, on the *entire* message base, all an application has to do is scan through this file!

This is the one and only reason why a personal mailscan on Hudson bases is so fast. Everyone has sometimes logged onto a QuickBBS or Remote Access (pre 2.00) board, and saw how fast it was...

* lastread.bbs

This file holds the lastread pointers for all users in all boards.

Advantages:

* Speed. Because all messages are stored in one set of files, an application has to open 1 set of files and can then reach each single message on the system without having to change directory, or open any other file.

For tossers, the speed can also be very high, because it is easy to buffer information. You're only doing sequential writes to a single set of files, which makes buffering relatively easy.

* It also takes less space than *.MSG, as it doesn't take a file for each message. Internally, the Hudson base works with 'records' of 256 bytes, however, so there is some internal 'slack'.

* Widespread and very popular. Lots of high quality support packages are available.

Disadvantages:

* Vulnerability:

- All messages are stored in one set of files. If these get corrupt, your entire message base can be lost.

- The headers are stored separately from the bodies. If the headers and bodies get 'out of sync', you won't be able to do a lot of useful reading.

* Limits. Hudson is limited to 200 areas, and 16 Mb of message text (total space taken by all message bodies). For a somewhat larger system, this is not enough anymore.

9.3 Squish

Squish could be placed somewhere between Hudson and *.MSG. It stores messages for every *area* in a set of files. So, messages for the same area go into the same set of files, but messages for other areas go in other files.

A Squish file area basically consists of 3 files:

* <name>.SQD

This is the 'datafile'. It contains the messages (both the headers and the bodies). Messages are stored in a 'doubly linked list' in the message area.

In every message area, there are two 'doubly linked lists': a list containing the 'active' messages, and a list containing the deleted messages.

The linked list of deleted messages can be used when a new message needs to be added: if the space that the deleted message once took is large enough, its space can be reused by a new message.

In addition, Squish can purge messages 'on-the-fly': when messages have to be added to the message area, Squish can delete messages while it is tossing in order to keep (no more than) a certain amount of messages in that area.

* <name>.SQI

This is an index file that holds, for each message, the offset in the .SQD file where that message starts, and the 'hash value' of the 'TO:' field of the message.

The hash value is a 32 bit number, that is calculated using the TO: name. Although this number is not entirely unique, I have not seen two real names with the same hash value.

The hash value can be used to do a quick personal mailscan: just calculate the hash value for the name of the user, and compare these values to the hash values found in the index.

* <name>.SQL

This file holds the lastread pointers for each user.

For the above, <name> is the 'base filename' that can be anything the user wants (and that is used to distinguish one area from another).

Advantages

- * Safe: because all areas are stored separately, the worst that can happen if a file gets corrupt, is loss of one area. Due to the 'linked list' structure, a damaged area can usually be (fully) repaired. Even from a badly damaged file, you usually lose no more than a handful of messages.
- * Efficient with space: it takes less space than *.MSG bases, and is comparable to Hudson in this respect.
- * Quite fast. It is faster than *.MSG, although not as fast as Hudson (it doesn't have to open a file for each message, but it

has to for each area.

Disadvantages

- * People that have always used Hudson, will find tossing and maintenance to be slower, especially if 'on-the-fly' purging is used.
*.MSG people will not suffer from this.
- * Less 'third party' support than Hudson.

For a long time, Squish was the best thing in my opinion: it was space efficient, sufficiently fast if 'on-the-fly' purging was turned off, and very safe.

9.4 JAM

JAM was only 'released' a short while ago (late 1993). It uses the same concept as Squish: messages are stored in one set of files per area. Messages are not stored using linked lists, however, and 'on-the-fly' purging is not part of the standard.

Every JAM area consists of the following files:

- * <name>.JHR

This file contains the headers of the messages, and the kludges.

- * <name>.JDT

This file contains the actual message text (the 'bodies' of the messages).

- * <name>.JDX

This is an index that is holding, for each message, the offset in the .JHR file where the header starts. It also contains a CRC value, calculated over the TO: field of the message. This can, just like the hash value for the Squish areas, be used for a fast personal mail scan.

- * <name>.JLR

This file holds the lastread pointers for all users.

Advantages:

- * Safe. It is much safer than Hudson (because of the separate storage for different areas, but not as safe as Squish (whose recovery capabilities are better, because of the 'linked list' design)).

- * Faster than *.MSG, slower than Hudson.
- * Very space efficient, better than Hudson and Squish.
- * Already is, or will be, supported by many authors of well known (and high quality) software packages that are available for Hudson bases.

Disadvantages:

- * It is quite new, and still has some weak points (in both design and (some!) offered implementations).
- * Hudson users will find it slow in both tossing and maintenance.

10. timEd configuration reference

This section will list all keywords that can be used in the timEd.cfg configuration file. The listing is alphabetical, making it easy to find the keyword you are looking for.

Included in the timEd package is a sample timEd.cfg, that lists all commands in a more logical order.

This reference section is provided to offer detailed information about every configuration keyword.

General rules for the configuration file:

The configuration file for timEd should be called timEd.cfg and must reside in the current directory, so timEd can find it (use the -C command line parameter otherwise).

It is a plain ASCII file and may contain comments. A comment is a line that starts with a semi-column, like this:

```
; this is a comment.
```

Blank lines are ignored, all others should contain a keyword. Usually, in this form:

```
<keyword> <value> [value value]
```

So first the keyword, followed by the value of that keyword, optionally followed by more values..

The following keywords can be used in timEd.cfg:

ADDRESS

Use this keyword to let timEd know what your address is. You can enter up to 35 of them, for your main address and AKA's.

The first one given will be the default address. If you want to use an AKA for a certain area, you must use the -P switch (see EchoArea keyword). -P switches found in a Squish configuration file will be recognized and used. Also AKA's found in other configuration files (like Fastecho.cfg) will be recognized and used.

However, you *must* always define all addresses to use in timEd.cfg. By playing with the order of the addresses (and/or by leaving some addresses out), you can influence the AKA-matching process.

Example:

```
address      2:281/527
address      2:500/133.999
```

AKAFORCE

You can force akamatching by using the AKAFORCE keyword.

Format:

```
AKAFORCE <mask> <address to use>
```

example:

```
AKAFORCE 50:*/*.* 49:500/1
```

This means: always use 49:500/1 as address when mail is sent to any zone 50 address. This forcing will then always be active (regardless of your 'AKAMATCHING' setting), and will take precedence over 'automatic' akamatching.

AKAmatching

Format: AKAmatching < Yes | No >

This option is interesting if you have more than 1 address. If you set this to 'Yes', timEd will automatically try to find the most appropriate address to use when writing a message.

Say, for example, that you have two addresses: 2:281/527 and 60:100/112.

If you write a message to 2:500/133, you probably want to use your 2:281/527 address.

If you write a message to 60:100/1, you probably want to use your 60:100/112 address.

In this case, timEd would try to find the address (AKA) that 'matches' the destination address best.

It first looks for a matching zone, and if more than one match is found, it'll try to find an address where both 'zone' and 'net' match. If there is still more than one match after that, it will just take the first match.

See the section about writing netmail for more info.

AREALISTSORT

Use this keyword to let timEd sort the arealist, on the

following criteria:

T - Type of area (order: Netmail, Local, Echomail).
N - Nametag ('areatag' of the area, like 'Artware').
D - Description of the area.
A - Aka used for that area.
G - Group of the area.

You can combine more than one criterium.

Example: AreaListSort TGN

This will sort on type, then on group, then on Nametag.

AREASBBS

This gives the path and filename of an Areas.bbs format file. Many echomail processors are able to generate such a file.

TimEd also recognizes Squish and JAM areas in this file (in addition to the 'standard' *.MSG and Hudson areas).

Squish areas must start the definition with a '\$', like:

```
$c:\msgs\tub          tub          2:281/500
```

JAM areas must start the definition with a '!', like:

```
!c:\msgs\jamdev      jamdev      2:281/500
```

ArcMailAttach

Format: ArcMailAttach < Yes | No >

This keyword is needed to support the 'Direct' message attribute (and other, extra attributes like 'Immediate', 'Kill file sent' etc) for people that are running FrontDoor (and maybe D'Bridge too..).

Frodo supports the 'FLAGS' kludge that makes these extra attributes available.

For Binkley-users (where Squish packs the netmail) one must set both 'Crash' and 'Hold' to get the 'Direct' attribute. The other attributes are simply not available in that environment.

How does timEd know what system you run? It looks in your Squish.cfg, if it finds 'Arcmailattach', you are running FD, otherwise Binkley.

If timEd doesn't read a Squish.cfg, you can force

'Arcmailattach' by specifying:

ArcMailAttach Yes

Anyone not using Squish (but Fastecho, for example) and running Frodo (or similar) should use this keyword.

Default is *not* ArcMailAttach.

CCVERBOSE

This can prevent timEd from expanding CC:'s into <name (address)> form with.

Possible values:

Yes: still expand CC:'s to 'name (address)' form.

No : only show to whom the original was addressed, and the number of other recipients.

Example:

CCVERBOSE Yes

CLOCK

A clock can be shown in the message reading and edit screen. In the DOS version, you can set this to 'Yes' to see a clock, while 'No' will give you the amount of free memory instead of the time. In OS/2, the clock is always active.

The DOS version will give up time slices to DESQview and Windows (or call INT 28 if neither of those are found), because updating the clock means it cannot be blocked on I/O anymore. With "Clock No" it will be blocked on I/O again, so that's the way to go in case of an unacceptable performance hit / unsupported multitasker.

In the OS/2 version, the clock runs in a separate thread that is sleeping most of its life, while the main thread is blocked on I/O, waiting for a keypress.

For those who do not understand what I wrote above: timEd is very multitasker friendly, and won't 'steal' any processor time away from other tasks that may be running on your system.

COLOURS

Lots of colours can be specified by the user. The numbers specify the colour to be used.

Note: by default, timEd uses a monochrome setup. So if you don't have a colour monitor, you can initially leave all color_... statements out (in fact, you can do that with a colour monitor as well, it just looks very boring :-).

The numbers are as follows:

	Foreground	Background
BLACK	0	0
BLUE	1	16
GREEN	2	32
CYAN	3	48
RED	4	64
MAGENTA	5	80
BROWN	6	96
LIGHTGRAY	7	112
DARKGRAY	8	n.a.
LIGHTBLUE	9	n.a.
LIGHTGREEN	10	n.a.
LIGHTCYAN	11	n.a.
LIGHTRED	12	n.a.
LIGHTMAGENTA	13	n.a.
YELLOW	14	n.a.
WHITE	15	n.a.
BLINK	128	n.a.

Now take the preferred colours of the foreground and the background of a certain item, and add them.

So, blue (foreground) on black (background) would be $1 + 0 = 1$

lightgray on black is $7 + 0 = 7$

white on blue is $15 + 16 = 31$

etc.

Tip: A much better way to define your colours for timEd is using Dick Pluim's TIMCOLOR utility. It will let you change timEd colours in an interactive way, and can write out configuration segments (to be included in timEd.cfg with the 'Include' keyword) with the corresponding COLOR_.. keywords.

Here are the colours you can define, and the keywords to use + my personal setting of these colours.

There are three main categories:

* color_as...

For the area selection screen, where you can select a message

area to read. These colours will also be used for the message header list (ALT-L) and the file selection box that will appear when you use a 'wildcard' in the subject when doing a file attach.

* color_msg...

For the part where you will spend most time: the message reading screen.

* color_pop...

For popup-screens, like help screens and errors.

Area selection screen settings

Area selection 'Title' Bar, at the top of the screen:

color_asbar 112

Area Selection Frame, the single line around the total screen:

color_asframe 4

Area Selection Normal Text, like the names of the areas:

color_astext 7

Area Selection Highlighted bar, the message area that is currently selected:

color_ashigh 31

Special colour, currently only used in a msg header list for personal messages:

color_asspecial 14

Message reading screen settings

Message reader Header, like the From: and To: field contents:

color_msgheader 7

Message reader date, part of the message header:

color_msgdate 14

Message reader 'labels', like the words 'TO:' and 'From:':

color_msgdata 15

Message reader reply link information:

color_mslinks 3

Message reader message attributes:

color_msgattribs 3

Message reader horiz. line, between header and text:

color_msgline 9

Message reader quotes, quoted text (starting like GvE>):

color_msgquote 14

Message reader normal text, the body of the message:

color_msgtext 7

Message reader 'Status' bar at bottom, with current msg area etc:

color_msgbar 31

Message reader origin, the origin of a message:

color_msgorigin 15

Message reader kludges (controlinfo like MSGID and SEEN-BY):

color_msgkludge 3

Message reader, the TO: field in the header for personal messages. Also used for 'highlights' in the message body when using the F)ind feature.

color_msgspecial 12

Popup boxes settings

Popup boxes frame, the box around a popup text:

color_popframe 121

Popup boxes text, the text inside the box:

color_poptext 112

Generic entry colour

Colour of all entry fields (for example when you edit a header, the field where you enter the TO: name):

color_entry 30

Internal editor colours

Current line, normal text (set to COLOR_MSGTEXT if you don't want the current line to be 'highlighted').

color_editcurnormal 15

Current line, quoted text (set to COLOR_MSGQUOTE if you don't want the current line to be 'highlighted').

color_editcurquote 6

Current line, text in a block (set to COLOR_EDITBLOCK if you don't want the current line to be 'highlighted').

color_editcurblock 113

Text that is in the currently defined block:

color_editblock 112

Colour of the 'Hard Return Token':

color_edithcr 8

Find Screen Colours

The colour of the frame of an active window:

color_findactive 11

The colour of the frame of an passive window:

color_findpassive 1

The colour of normal text (search strings, and 'lables' like 'Action:')

color_findtext 7

The colour of data-items (value of a label, 'Read' as the value of the 'action' parameter for example):

color_finddata 3

The colour of the highlighted ('selected') entry (highlighted bar):

```
color_findhigh 113
```

Confirm_Edit_Exit <Yes | No>

This controls whether timEd asks for confirmation when you press <ESC> in the internal editor (to abort the message). If you set this to 'Yes', timEd will first ask if you're sure, before it dumps all the text you so carefully edited :-)

Confirm_Exit

Format: Confirm_Exit <Yes | No>

This controls whether timEd asks for confirmation before exiting to DOS. If you set this to 'Yes', timEd will ask if you are really sure, before exiting to DOS.

Confirm_Delete

Format: Confirm_Delete <Yes | No>

This controls whether timEd asks for confirmation before deleting a message. If you set this to 'Yes', timEd will ask if you are really sure, before deleting a message.

ECHOAREA, NETAREA

Area definition, Squish alike. You can specify areas in timEd.cfg as well, in addition to the areas found in Squish.cfg.

The squish.cfg file is read *after* these are read, however, so you can 'overrule' areas here (areas already defined in timEd.cfg will be skipped in squish.cfg and/or areas.bbs).

This is the only place where you can give a 'long' description of an area (can't do that in squish.cfg, of course) and where you can specify 'default message attributes' for newly created messages in that area.

It looks quite a bit like definitions in squish.cfg. There are 3 keywords:

- EchoArea - for echomail areas
- NetArea - for netmail areas (define as many as you want)
- LocalArea - for local areas

The format for all of these:

```
<keyword> <description> <areatag> <location> [-Pxxx -Axx -$ -J -H]
```

```
<keyword>          is NetArea, EchoArea or LocalArea.
<description>     is a description of the area, between double
                   quotes
<areatag>         is the official areatag of the area (like: TUB).
                   Even if the area is a Local area or a Netmail
                   area, the area must have a UNIQUE areatag, as
                   timEd uses these to distinguish areas!
<location>       is the directory/basename/board where the area
                   resides.
```

And where:

The default is *.MSG format, but:

-\$ is for squish style areas.

-J is for JAM style areas.

-H is for Hudson base areas.

And:

-A gives default attributes.

```
  p = private,
  c = crash,
  k = killsent,
  d = direct,
  i = immediate,
  r = readonly (seen notes below);
```

-Pz:nnn/nnn.p is the AKA to use for this area (also active for NetArea). The default is to use the primary address.

You can make an area 'read-only' if you add -Ar in the area definition in timEd.cfg. The term 'Read-Only' is not really correct, but replying to a message in that area or creating a new one is not possible. It is meant as a convenience (for example to prevent you from writing messages in a 'personal mail area') and certainly not as a security measure! Deleting a message, for example, is possible. So is mass copying/moving to such an area.

Examples:

```
NetArea "Netmail Area" netmail c:\fd\netmail -Apk
```

My primary netmail area, with a nonsense areatag, is in *.MSG format and all messages get the 'private' and 'kill/sent' bits

by default. The 'local' bit is always added to all messages as well.

```
NetArea "Wlink net" w_net c:\fd\wnet -Ap -$ -P60:100/112
```

A second netmail area, in Squish format. Messages get the 'private' bit by default and I use my AKA 60:100/112 in this netmail area.

```
LocalArea "Bad_Msgs" BAD_MSGS C:\Squish\Msgs\Bad -$
LocalArea "Dupes" DUPES C:\Squish\Msgs\Dupes -$
```

Here I add my bad_msgs and dupe areas. I like to see what's happening.

```
LocalArea "SysOp" Sysop d:\local\sysop -$ -Ap
```

I define my SysOp area, default private messages.

```
EchoArea "JAM development" JAMDEV d:\echo\jamdev -J
```

An echomail area in JAM format.

```
EchoArea "Contrast" CONTRAST d:\echo\CONTRAST -$
```

An echomail area (my local points area, defined here as well as in squish.cfg, but I define it here, because I want it at the top of my area selection screen..)

```
EchoArea "Points" POINTS.133 c:\sqsh\pnt -P2:500/133.999
```

Another echomail area, where I use my AKA 2:500/133.999, *.MSG format.

```
EchoArea "Artware testers" artware.closed 3 -H
```

An echomail area, in the Hudson base format (that's what the -H is for). The area is board number 3 in the Hudson base.

ECHOLOG

Where to put the echotoss.log file (name + path). If this is specified, an echotoss.log file will be written when you exit timEd.

An echotoss.log file is a file containing a list with the

areatags of areas that contain newly entered messages. You can pass this file to your tosser/scanner to scan out and pack these new messages. (For Squish, this is done by using the '-f' command line parameter).

Example:

```
EchoLog C:\timEd\echotoss.log
```

EDITOR

This tells timEd what editor you want to use to edit your messages.

There are two possibilities:

'Internal' - this will let you use timEd's internal editor.

'<path + filename>' - if you give the name of a program, timEd will execute that program instead of using timEd's internal editor. Whenever it is time to edit a message, timEd will execute this program. Entering a full path is not required (because timEd will look for the editor in your PATH), but speeds up things.

TimEd will call the editor with 1 parameter: the name of the temporary file that timEd created and that contains the message body to be edited.

You can also use a batchfile here. You could use this to start up your editor with extra options, or to call other programs before or after the editor (a spell-checker maybe?).

TimEd calls batchfiles with two parameters: the name of the file to edit (%1) and the areatag (%2). You might use the areatag to perform some magic in a batchfile, like calling different editors for different areas.

Example:

```
Editor c:\misc\q.exe  
or  
Editor Internal
```

Empty_Tearline

Format: Empty_Tearline <Yes|No>

This controls whether echomail messages will have a tearline mentioning the timEd version (like: --- timEd/B10) or an empty tearline.

If you set this to 'Yes' an empty tearline will be created. In

that case, a ^APID kludge will be added to show the timEd version you are using.

End_of_Area_Menu

Format: End_of_Area_Menu <yes | No>

If this is set to 'Yes', and timEd reaches the last message in an area, pressing <right> or <enter> will have a message pop up, that gives you the choice to:

- go to the next area with new mail (<right>, <enter>, '+').
- exit the area and go to the Area Selection Screen (<esc>).
- just stay where you area (any other key).

Lazy readers can just keep on pressing <right> or <enter>, and cycle through all their areas.... :-)

With this set to 'No' timEd will just 'beep' at the end of an area, and do nothing else.

ENTERMARKS

If this is set to 'Yes', the <enter> key will mark a message. If this is set to 'No', <enter> will not mark a message but will function as <cursor right> and move you to the next message. In this case, <space> will mark messages.

Example:

EnterMarks No

FASTECHOCFG

TimEd can read the Fastecho.cfg for you (current version: 1.30, 1.41), and get all echomail areas from that configuration. Note that Netmail, bad_msgs and dupes are not read by default; add 'ReadNetAreas Yes' and/or 'ReadLocalAreas Yes' to your timEd.cfg file to let timEd read those areas as well. Different AKA's used in your Fastecho setup will be recognized and used when creating a message.

Give full path and name of the file.

FastechoCFG c:\fe\fastecho.cfg

FDNODELIST

This tells timEd where it can find a FrontDoor nodelist (used by.... FrontDoor! Yes!)

TimEd can use the files called USERNAME.FDX, NODELIST.FDX, NODELIST.xxx, FDNODE.FDA, FDNET.PVT and FDPOINT.PVT (the last three only if applicable, not required).

When looking up a SysOp-name, timEd will try to locate *all* matches for the (part of the) name entered. So, if you enter 'Essen', it might find both 'Gerard van Essen' and 'Martin Essenburg'.

With the frontdoor nodelist, you can always get the same SysOp name more than once, and pick one of their addresses (HUBs etc. often have more than one address). Unlike the Version 7 nodelist, duplicate SysOp names are always preserved in the index.

Example:

```
FDnodelist      C:\Frodo\Nodelist
```

FMAILCFG

TimEd can read the Fmail config files for you (current version: 0.96), and get all echomail areas from that configuration. Note that Netmail, bad_msgs and dupes are not read by default; add 'ReadNetAreas Yes' and/or 'ReadLocalAreas Yes' to your timEd.cfg file to let timEd read those areas as well.

Different AKA's used for message areas in your Fmail setup will be recognized and used when creating a message.

Give full path and name of the file Fmail.cfg

```
FmailCFG c:\fmail\fmail.cfg
```

FOLLOWHELLO

See 'HELLO'.

FREQATTRIBUTES

This is the default status of a file request message created by the ctrl-f function.

Values: i)mmediate, c)rash, d)irect, h)old, k)ill.

Example:

FreqAttributes ck

File request messages created by timEd's 'ctrl-f' function will now automatically have the 'crash' and 'kill/sent' attributes.

GECHOCFG

! Note: older versions of GEcho (older than 1.01) are not supported!

TimEd can read the GEcho config files for you (current version: 1.02), and get all echomail areas from that configuration. Note that Netmail, bad_msgs and dupes are not read by default; add 'ReadNetAreas Yes' and/or 'ReadLocalAreas Yes' to your timEd.cfg file to let timEd read those areas as well.

Different AKA's used for message areas in your GEcho setup will be recognized and used when creating a message.

Give full path and name of the file Setup.ge

GEchoCFG c:\GEcho\Setup.ge

HELLO, REPHELLO, FOLLOWHELLO

HELLO : This defines the string to start a new message with. You usually say 'hello' before you start a message, so this is called 'hello' and will be referred to as 'hellostring' throughout the documentation.

REPHELLO : This defines the string to start a reply with. This is usually something like: 'In a message xxx wrote to yyy'. This will be referred to as the 'rephello string'.

FOLLOWHELLO : This defines the string to start what to start a 'follow-up' (created by pressing ALT-O) message with.

Always put the 'hello', 'followhello' and 'rephello' strings between double quotes.

There are several 'variables' that can be used in this string (and in the 'rephello' and 'followhello' strings as well):

%to : The full name of the person that the message IS addressed to (for a new message) or that the original WAS addressed to (for a reply, the REPHELLO keyword).

%fto : As %to, but only the first name of that person.

%from : The full name of the person who wrote the original message (in the case of a reply, so the REPHELLO keyword). In the case of a new message this would be the name of the person who wrote (writes) this new message (but that, of course, is you yourself :-)

%ffrom : As %from, but only the first name of that person.

%subj : Subject of the message

%orig : Address of the sender of the message (like 2:281/527)

%dest : Address of the recipient of the message (like 2:281/527)

%time : Time the message was written (like 01:25)

%year : The year the message was written (like 1993)

%mon : The month the message was written (like jan, feb etc)

%day : The day of the month msg was written (a number)

%dow : The 'day of week' msg was written (like mon, tue, wed etc)

An example to clear up things:

Let's say I'm writing a message to Scott Dudley, so that's a new message, and the 'HELLO' keyword would be of significance here. The header would look like this:

```
From: Gerard van.Essen
To : Scott Dudley
Subj: Your MSGAPI
```

In this case, %to == 'Scott Dudley' and %fto == 'Scott'.

If my timEd.cfg has:

```
hello          "Hello %to!"
```

My message would be started with:

```
Hello Scott Dudley!
```

Of course, because we treat everyone like our best friend in Fidonet, I have:

```
hello          "Hello %fto!"
```

As a result, my message will start with:

Hello Scott!

Right, that's the 'hellostring'. Now, let's say Scott receives my message, and decides to write something back. This will be a reply, so the 'rephello string' will be used.

Looking at our message again (the same one, as this is the message Scott is answering..):

```
From: Gerard van.Essen
To  : Scott Dudley
Subj: Your MSGAPI
```

In this case, %to == 'Scott Dudley' and %fto == 'Scott', %from == 'Gerard van.Essen' and %ffrom == 'Gerard'

Now if Scott uses timEd, *and* he has this in his timed.cfg:

```
rephello      "%from wrote in a message to %to:"
```

His reply would start with:

Gerard van.Essen wrote in a message to Scott Dudley:

And he can start typing..

HUDSON_OFFSET

Multi-user support for lastread pointers in Hudson bases as well: use the Hudson_Offset keyword to specify offset in lastread file. First user (and default) is offset 0.

If you use the same base with more than one user, give one user Hudson_Offset 0, someone else Offset 1 etc.

Be aware, though, that BBS systems also use space in the lastread file. Make sure that your setting does not clash with the offset of some BBS user!

Example:

```
Hudson_Offset 2
```

HUDSONPATH

This will tell timEd where your Hudson base is located. If you don't have a Hudson base, this keyword is not needed (in fact, in that case it is even better if you leave it out!).

In this directory, timEd will also create the ECHOMAIL/NETMAIL.BBS files, that will signal to the tosser which new messages were entered. With this file, the scanner can very quickly export new messages.

Example:

```
Hudsonpath c:\fd\msgbase
```

IMAILCFG

TimEd can read the Imail config files for you (current version: 1.50), and get all echomail areas from that configuration. Note that Netmail, bad_msgs and dupes are not read by default; add 'ReadNetAreas Yes' and/or 'ReadLocalAreas Yes' to your timEd.cfg file to let timEd read those areas as well.

Different AKA's used for message areas in your Imail setup will be recognized and used when creating a message.

Give full path and name of the file Imail.cf.

```
ImailCFG c:\imail\imail.cf
```

INCLUDE

With this keyword, you can include another file in the config.

If you like to change between different coloursets, for example, you can easily switch between them by putting them into two files (like set1.col and set2.col) and putting

```
'Include c:\timed\set1.col'
```

or

```
'Include c:\timed\set2.col'
```

in your config.

This feature comes in very handy when you use Dick Pluim's colour configuration program for timEd!

Format:

```
Include <path+filename>
```

Example:

```
Include c:\timed\myareas.lst
```

INTLFORCE

Format: INTLFORCE <Yes|No>.

With this set to 'yes', *all* netmail messages generated by timEd will have an INTL kludge (instead of only inter-zone messages).

In some environments this seems to be necessary. Do not use this unless you really need it!

JAMLOG

Where to put ECHOMAIL.JAM and NETMAIL.JAM files. If you have any JAM echomail/netmail areas, these files will signal to the tosser which new messages were entered. With this file, the scanner can very quickly export new messages.

With the current version of RA (2.01) and Fastecho (1.30), this file should be put in the directory that also holds the Hudson base.

If you leave out this keyword, but have JAM areas anyway, the JAM areas where new mail was entered will be added to your echotoss.log file. This is useful for tossers that do not support the special echomail/netmail.jam files (like Imail).

Example:

```
Jamlog c:\fastecho\msgbase\
```

JAM_ZONEGATE_BIT

This controls whether or not timEd uses the 'MSG_GATE' bit in JAM netmail areas (for a zonegated message).

If your tosser supports it (at the time of writing, it is known that GEcho 1.11 will support it), set this to yes.

What happens is that timEd will just set this bit in the JAM message (and do nothing else with it), if you tell timEd to make an inter-zone message zonegated. The tosser will recognize this bit, and do the re-addressing of the message to the zonegate etc.

If you set this to 'No', timEd will use another method to make the message zonegated. Chances are, your tosser will work OK

with this (at this time, Fastecho 1.40 is an example of such a tosser), but this is definitely not guaranteed, as this method is quite strange (due to the JAM structure and timEd's implementation of it).

JUMPY_EDIT

This keyword controls timEd's behaviour when you have to edit one field (i.e.: not the internal editor!). For example, when you want to write a message to file, you have to give the filename. You can do that in an entry field, where you can edit the filename to write to. This keyword controls timEd's behaviour in such fields.

When editing a field with a default already filled in, timEd can now clear the string when the first key you press is a normal character (and not a movement key, for example). This is what Qedit does, as do many setup programs (FDsetup, Gsetup etc).

I consider this to be 'jumpy' behaviour (where the h*ll is my default string?), so it is controlled by a this keyword:

Jumpy_Edit <Yes|No>.

If you set this to 'Yes', the field will be cleared if you press a non-edit key as the first key.

Please note, that if you set this to 'No', you can always use CTRL-Y to clear the entire field in one stroke!

Kill_Original.

Format: Kill_Original < Yes | No | Ask >

This controls whether you want timEd to delete a message, after you wrote a reply to it. (This is only functional in the netmail area). You can use this to keep your netmail area clean.

LASTREAD

TimEd supports multiple lastread pointers. This is necessary if you are not the only one using timEd on 1 machine (as the lastread pointers would have to be shared by 2 (or more) users then).

If you are the only one using the message base, or if you are the SysOp (and are consequently occupying the first position in the leastread file) you do not need this keyword.

This is usually the case, so most people don't need this keyword at all.

With this keyword, you can give the name of the lastread file for *.MSG areas. The default is "lastread".

Example:

```
LastRead lastreal
```

LOCALAREA

See 'ECHOAREA'.

LOWLEVELKB

This keyword can switch on the 'low level' keyboard routines. With these routines, timEd will recognize the ASCII 224 character. In addition, timEd will ignore any keyboard redefinitions caused by ANSI-driver keyboard programming. The disadvantage is, that it may cause some TSR's to stop working (or even lock up the machine!). Some people also reported random lockups using these routines, and other weird problems (timEd waiting for a keypress during a mass-move in the List mode, for example).

On the author's machines, the routines never gave any problems, though. If you have strange problems, try to switch this off.

Values can be 'Yes' (use Low Level routines, the default), or 'No' (don't use).

MACRO

Macros can save you typing names and addresses of people you regularly send mail to.

It works like this: in a netmail message, you enter a (short) name, that was also listed in timed.cfg using a 'macro' statement. TimEd will recognize this statement and expand it for you, saving you a lot of typing. This may include the 'to:' field, the address the message is sent to, and the subject.

Examples:

```
macro am,areamgr,2:281/520,my_pass
```

When you enter 'am' in the 'to:' field, timEd will replace it with 'areamgr', and fill in the address (2:281/520) and subject (my_pass) lines for you. So, a total of 3 items were listed after the macro name.

macro et,Erik Troost,2:281/527.2

This will expand to 'Erik Troost' with the address filled in. That's 2 items after the macro name.

macro rvdn,ROn van der Nagel

This will only expand the macro (rvdn) to ROn van der Nagel. Only 1 item listed behind the macro name.

Note that spaces are allowed in a macro statement. They are never stripped, so don't put them where you don't want them!

Also note that timEd doesn't default to anything when a macro is used, as far as addresses are concerned. So you should always give full 4D addresses in macros (so **not** "macro et,Erik Troost,281/527.2", because that will lead to the address 0:281/527.2!).

Move_Notes

Format: Move_Notes <Yes|No>.

This controls whether timEd will put "Moved / Copied (from: xxx) by xxxxx using timEd" at the top of a message that was Moved or Copied using ALT-M.

Please note that FORWARDED messages will always have the message.

Messages Copied or Moved using the L)ist mode will never have the message at the top.

NAME

Type your name (and optionally aliases, just put in more NAME keywords) here. The NAME will be used in the FROM: field of every message you write.

The aliases will be used as extra names to look for in the personal mail scan (ALT-P from the area selection menu, {+}).

While in a message area, you can also pick one of your names to write messages with. Press CTRL-N to change the username from within timEd. The default is to use the first NAME found in timEd.cfg.

Always put the name between double quotes.

Example:

name "Gerard van.Essen"

name "Gerard van Essen"

NETAREA

See 'ECHOAREA'.

NETMAIL_SEMAPHORE

Format: Netmail_Semaphore <path+filename>

This can be used to configure a file to be created (or touched) when netmail is generated. For frontdoor this should be fdrescan.now in the frontdoor directory.

You can use this file to give a signal to another program (usually your mailer) that your netmail area changed. That program can then rescan the netmail area.

Example:

Netmail_Semaphore c:\fd\fdrescan.now

Netmail_Tearline

Format: Netmail_Tearline <Yes|no>.

This controls whether netmail messages will have a tearline or not. Tearlines are not required in netmail messages, although some programs (usually areafix alike programs) use it to find the end of the message body.

NODELIST

This tells timEd where it can find a Version 7 nodelist (used by BinkleyTerm and Portal of Power, for example).

The nodelist is very useful when entering netmail: timEd can look for addresses and SysOp names in the nodelist, to find the corresponding data of that node. (So, when entering a name, timEd will try to find the address, when entering an address, timEd will try to find the SysOp's name..).

The Version 7 nodelist can be produced by several programs, like Xlaxnode, Qnode and Fastlist. TimEd needs the files called SYSOP.NDX, NODEX.NDX and NODEX.DAT.

When looking up a SysOp-name, timEd will try to locate *all* matches for the (part of the) name entered. So, if you enter 'Essen', it might find both 'Gerard van Essen' and 'Martin Essenburg'.

If you use Qnode, and put 'Keep Allusers' in qnode.cfg, you can even get the same SysOp more than once, and pick one of their addresses (HUBs etc. often have more than one address). Recent versions of 'Fastlist' have the same capability.

Example:

```
nodelist      C:\Binkley\Nodelist
```

ORIGIN

This tells timEd what the default origin is. This line will appear at the bottom of every echomail message you write.

You can override this default on a per area basis (by pressing ALT-H in timEd, while you are in a message area).

Always put the origin between double quotes.

```
origin      "   Contrast BBS, 070-3234903 [V22..32bis]   "
```

Personal_Skip_Received.

Format: Personal_Skip_Received <Yes | No>

This is for the personal mailsan (ALT-P).

When this is set to 'Yes', personal messages that have the 'received' status will be skipped by the personal mailsan (so you only see unread messages).

When this is set to 'No', the personal mailsan will also show messages with the received status.

Please note that the personal mailsan always starts scanning at the lastread pointer, regardless of this setting.

{+} The personal mailsan is only functional in registered versions.

PRINTER

Here you can tell timEd what port your printer is attached to. When using ALT-P from the message reading or list screen, timEd will send the message to what is listed here. The default (if you don't put this in timEd.cfg) is LPT1.

Note for Novell Lite users: NWLite seems to be unable to catch timEd's printer output. However, if you specify 'stdprn' as the printerport, NWLite *will* catch it.

Example: Printer LPT2

READLOCALAREAS

This keyword can let timEd read areas marked as local areas from your tosser configuration files.

If you set this to 'Yes', timEd will read those areas, if this is set to 'No', local areas will be ignored and will not show up in the arealist.

READNETAREAS

This keyword can let timEd read areas marked as netmail areas from your tosser configuration files.

If you set this to 'Yes', timEd will read those areas, if this is set to 'No', netmail areas will be ignored and will not show up in the arealist.

REPHELLO

See 'HELLO'.

Show_Edit_HCR

This defines whether timEd will show the 'hard returns' in the internal editor. If this is set to 'Yes', timEd will show a special symbol at the end of every line that has a 'hard return'.

This is only the startup value, when you are in the internal editor, you can toggle this setting with ALT-H!

Example:

```
Show_Edit_HCR Yes
```

SHOWKLUDES

This tells timEd if you want to see the control information in the messages by default (or not..).

For this purpose, 'kludges' includes real kludges (like MSGID, REPLY, PID, PATH etc.) but also SEEN-BY lines.

This is the setting at startup, you can toggle this setting from within timEd by pressing ALT-K or ALT-V when reading a message.

Example:

```
Showkludges Yes
```

SIGNOFF

This tells timEd what should be placed at the end of every message. Most people always use the same 'signoff', so why not let timEd do it for you..
Always put the 'signoff' message between double quotes.

```
Signoff      "Groetjes, .....   === Art ==="
```

In addition, you can also use \n in a string, to indicate a newline. So:

```
signoff "Greetings,\n          Gerard"
```

expands to something like:

```
--
```

```
Greetings,  
          Gerard
```

```
--
```

Note, however, that strings longer than +/- 70 characters cannot be edited from within timEd (with ALT-H).

SQUISHCFG

TimEd can read the Squish.cfg (current version: 1.10) for you, and get all EchoArea lines from that configuration. Note that that Netmail, bad_msgs and dupes are not read by default; add 'ReadNetAreas Yes' and/or 'ReadLocalAreas Yes' to your timEd.cfg file to let timEd read those areas as well.

-P switches found in Squish.cfg will be recognized and used when creating a message.

Give full path and name of the file. If it contains an AreasBBS statement, the areas.bbs file will be read as well.

```
Squishcfg c:\squish\squish.cfg
```

Squish_Offset

This is the equivalent of 'LastRead', but for Squish style areas.

You must give a number, that is the position of the lastread pointer in the *.SQL file (for Squish areas). For the first user this is 0 (which is the default). If you are the second user you should set this to 1, if you're the third user you should set it to 2, etc.

Again: most people do **not** need this.

Example:

```
Squish_Offset 1
```

This would let timEd use the second lastread pointer in the .SQL files (Squish_Offset 0 would be the first, which is the default).

Startup_Lines

This controls the videomode, the number of rows on the screen. This mode will be set on startup of timEd. The values that are allowed depend on the timEd version and mode:

DOS: 25, 28, 50.

OS/2 fullscreen: try it and you'll know (depends on videocard), in addition to the above, 30 and 40 or 45 may work.

OS/2 window: anything from 25-60 AFAIK (different fonts).

STARTUP_MODE

This determines the default mode ('view') timEd will have for showing your areas (Area Selection Screen). The mode can always be changed from the Area Selection Screen by pressing ALT-M, but this keyword sets the default at startup.

Use either 'All' (show all areas), 'New' (show only areas with new mail) or 'Tagged' (show only tagged areas) or 'NewTagged' (show only tagged areas that hold new mail).

Example:

```
Startup_Mode NewTagged
```

STARTUP_SCAN

This determines whether the message areas will be scanned for new mail at startup (value: Yes or No) or even for personal mail (value: Personal {+}).

So, use either 'Yes', 'No' or 'Personal'.

Example:

```
Startup_Scan Yes
```

SWAP_ON_EDIT, SWAP_ON_SHELL

These two statements determine timEd's behaviour when it calls other programs.

There are two options:

Yes : Swap timEd out of main memory when calling the other program. This will only leave about 250 bytes of timEd in main memory, thus making available a lot of memory for the other program to run in.

No : Don't swap timEd out, load the other program in the memory remaining, with timEd still in memory.

The second option takes up more memory, but is faster (no swapping has to take place). TimEd will be swapped to XMS, EMS or to Disk if no XMS/EMS is available. Swapping to disk is quite slow, of course..

Swap_on_Edit: Determines whether timEd is swapped out when the editor is called to edit a message. This is done often, of course (especially if you write a lot of mail :-), so if you have enough memory to hold both timEd and the editor in memory (and this is usually the case), I would set this to 'No'.

Swap_on_Shell: Determines whether timEd is swapped out when shelling to DOS, and when executing an external program by using the ALT-U feature (message base maintenance). Usually speed is not really important here, while lots of available memory can be very useful, so it might be a nice idea to set this to 'Yes'.

USENETMACRO

There is also another type of macro's: USENETMACRO. This works very similar to a 'normal' macro, but you can also supply a 'usenet style' name, that will be placed in the body of the message (TO: art@beard.next.nl for example).

The format is as follows:

```
USENETMACRO macro,Usenet_address,toname,[toaddress],[subject]
```

The last two items are optional.

Note that spaces are allowed in a macro statement. They are

never stripped, so don't put them were you don't want them!

Usenet_address should be an address like art@beard.nest.nl.
Toname should be the name to be filled in in the TO: field.
Usually,
this will be 'postmaster' or 'uucp'. Toaddress is the address to
send the message to, usually the address of your Usenet <->
Fidonet gate.

An example:

My Usenet <-> Fidonet gate is 60:60/0, and I want to send a
message to art@beard.nest.nl (why would I send a message to
myself? :-).

A macro could be:

```
Usenetmacro au,art@beard.nest.nl,postmaster,60:60/0
```

When entering a message, I can now just type 'au' in the TO:
field, and timEd will automatically fill in 'postmaster' in the
TO: field, 60:60/0 in the to-address field, and leave me at the
subject field.

When I start to edit the body, 'TO: art@beard.nest.nl' will be
placed at the top of the message.

WRITENAME

Here you can give the default filename the ALT-W (write message
to file) function will write to. When actually writing to a
file, you can of course edit this to be anything you want (this
is just the default). You can also put PRN (or similar) in here,
to write to a printer.

Example:

```
WriteName c:\timed\timeout.txt
```

WTRGATECFG

TimEd can read the Watergate area file for you, and get all
echomail areas from that configuration. Note that Netmail,
bad_msgs and dupes are not read by default; add 'ReadNetAreas
Yes' and/or 'ReadLocalAreas Yes' to your timEd.cfg file to let
timEd read those areas as well.

Different AKA's used for message areas in your WaterGate setup
will be recognized and used when creating a message.

Give full path and name of the main config file.

XMAILCFG

timEd can read the xMail area file for you (current version: 1.00), and get all echomail areas from that configuration. Note that Netmail, bad_msgs and dupes are not read by default; add 'ReadNetAreas Yes' and/or 'ReadLocalAreas Yes' to your timEd.cfg file to let timEd read those areas as well.

Different AKA's used for message areas in your xMail setup will be recognized and used when creating a message.

Give full path and name of the file AREAS.XM

```
xMailCFG c:\xmail\areas.xm
```

ZoneGate

Format: ZoneGate <Yes | No | Ask>

It works as one would expect: when this is set to 'No' timEd won't do any zonegating, if it is set to 'Yes' it will zonegate all inter-zone messages, and if it is set to 'Ask', timEd will ask whether it should zonegate or not..

If you use Binkley with Squish, you can also let Squish handle it (that's what I do).

Whether or not you need zone-gating depends on the software used by the node that actually sends the messages from one zone to another. Check with your Boss, HUB, Host or who-ever knows.

11. Miscellaneous notes.

11.1 Multitasking considerations.

Screen handling.

DOS version: TimEd uses direct screen writes (either to the physical screen, or to the DESQview virtual screen). There is no special code to prevent 'snow' on older CGA displays.

OS/2 version: timEd uses the VIO subsystem for screen writes.

Performance.

DOS:

TimEd will give up timeslices to DESQview, Windows and the OS/2 DOS box (although it would be much better to use the OS/2 version), if applicable. This means timEd will have no noticeable impact on system performance when it is idle (which is most of the time).

OS/2:

What? This is an OS/2 app! Of course it doesn't suck time from other tasks!!

File locking/sharing.

DOS:

File locking is only active if timEd successfully locked a file in the timEd directory (either the current dir or the one specified by the -c command-line switch).

OS/2:

File locking/sharing is always active.

11.2 Some things to remember.

- With the OS/2 version, spawning E.EXE or EPM.EXE directly from timEd/2 doesn't work. Using a batchfile (with "E.EXE %1" as the only contents) works OK...

11.3 Technical notes.

TimEd is written using Borland C++ 3.1 (DOS version) and Watcom C/C++ v10.10 (the OS/2 version), and makes extensive use of Scott Dudley's MSGAPI (thanks Scott!). It also uses small parts of the JAM message base API (made by Mats Wallin, Joaquim Homrighausen, Andrew Milner and Mats Birch).

It was developed on a 10 Mhz 286 (most of the time), and for a while on a 386-40 running OS/2 (also running my BBS) but even OS/2 couldn't protect the BBS from my programming skills (...), causing lockups and even reboots, so I went back to the 286.. Recently, development has taken place on a 386-40 again (running OS/2).

11.4 TimEd source code

Not available.

12. Frequently asked questions

Q1.

I wrote a nicely formatted message with my external editor. However, after timEd read the message in, the message looked quite different and the nice format was gone.

A.

In order to stay within the Fidonet specifications, timEd has to 'intelligently' strip 'hard returns' from your messages, before writing them to the message base. Usually, this works quite well and you won't notice it.

Sometimes it goes wrong, and for these cases you can instruct timEd to temporarily NOT strip these 'hard returns'. Look in timEd.prn ('Entering messages with timEd') for more details.

Q2.

In timEd.cfg I can define 'hellostrings' and an Origin. But for different areas, I want to use different origins. Is that possible?

A.

Yes, that's possible! When you press ALT-H while you are in a message area, you can define custom 'hellostrings' etc. for use in that area only! Using this, you can have different 'hellostrings' and origins for every area! See timEd.prn for more info.

Please note that permanently SAVING these strings is only available in registered versions.

Q3.

In timEd.cfg, I changed the 'Hellostring'. However, in ONE area, the old 'hellostring' is still being used! How come?

A.

You probably used ALT-H to edit the 'Hellostring' for that area. After using ALT-H, you can save {+} this information (in that area). That custom hellostring will then be used for messages in that area.

If you change the hellostring in timEd.cfg, that will not affect the custom 'hellostring' for that area.

See timEd.prn for more info about the ALT-H function.

Q4.

Will you implement support for 'Taglines' and 'Cookies' in timEd?

A.

No, I don't like to support automatically generated echomail pollution.

--

Q5.

I do a nodelist lookup on a SysOp that has several addresses (AKA's). However, timEd only shows me one of those addresses, and I *know* he has many of them.

A.

You are most likely using a Version 7 nodelist, and the compiler you used to generate it either doesn't support 'duplicate sysop names' in the nodelist indexes, or it was configured to strip duplicate names.

Currently, I know that Qnode and Fastlist (go for Fastlist!) can put duplicate sysop names in the index. Xlax does not have this ability.

This is not a bug in timEd, but a shortcoming in the nodelist compiler, or in your configuration of that tosser.

13. Thanks to..

TimEd is the result of enthusiastic efforts by many, many people. I know it sounds boring, but there are too many people to thank.

The whole thing was started by Scott Dudley, who released the MSGAPI code I wanted to play with, and the Squish message base format that I wanted to write a reader for..

Jelle Boot was the first one to actually use timEd apart from myself. He encouraged me to go on with the program. Even though he couldn't change any colours! But he had a nice program (a TSR that changed screen colours 'on-the-fly') to do just that :-)

Since that time, many people joined in. The current beta team consists of more than 30 people.

I owe them.

Apart from that there are very helpful registration, distribution and support sites.

I owe them, too.

This section, however, would not be complete without mentioning Evin Mulron, whose enthusiasm often inspired me to put in an extra hour of programming. Thanks!

I wouldn't try to put the names of all people that helped me in here. There are too many, and I would forget several of them. Everyone who helped me out, *knows* I am thankful to them. But, just to be sure:

T H A N K S !

"I couldn't have done it without you guys!"

14. Contacting the author.

You can reach me in several ways:

The best way:

Try to get the international ARTWARE conference. You will find me there, and several other experts :-)

My BBS:

Contrast BBS
+31-70-3234903
US Robotics V34, V.FC, HST 16k8, V32 terbo.

FidoNet: 2:281/527

Internet: art@cnh.wlink.nl

Or just send me a letter, or postcard:

Gerard van Essen
Leharstraat 67
2551 LB Den Haag
The Netherlands, Europe