

=====
timEd/beta

(C) 1992-'94 Gerard van Essen (2:281/527)

A message editor for Squish, *.MSG and JAM and Hudson
=====

timEd version : Gamma 1
Document last updated : feb 05 1994

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

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 DVAAWARE routines (DOS version).
- ! This product uses the JAM(mbp) API - Copyright 1993 Joaquim Homrighausen, Andrew Milner, Mats Birch, Mats Wallin. ALL RIGHTS RESERVED.

(Warning: this part doesn't contain any useful info, 'quick starters' should skip this.. :-)

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.

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).

Beta!

Right! This is beta software (and apart from that: *I* wrote it! :-), so be careful!

Before you run it, back up anything that is worth saving, make your will and warn your family and relatives.

There are no guarentees that timEd doesn't do anything terrible to your setup. Don't say I didn't warn you!

This documentation is also beta :-) I'm sure it lacks lots of info you are looking for, but will never find.

- * "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 not REQUIRED that you register the program. However, if you can afford to pay the registration for timEd (and in my opinion, 95% of the users can afford it), I suggest you register it anyway.

I would very much appreciate it, 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.

Finally: I know some people that do not want to register software that is still in the beta stages. Although the main reason that timEd is still in beta, is the fact that it didn't have an internal editor (and I believe a message editor should have one) until beta 11, and not because I think the code is unstable, I will respect this principle. If you really feel this way, you can hold off registration of timEd until version 1.00 is released (I expect this to be early 1994).

- * For COMMERCIAL use of the program, you should always register the program. You are, however, granted an evaluation period of 30 days. After these 30 days, you must either register the program, or stop using it. Using an unregistered version of timEd in a commercial environment for more than 30 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.

Installing timEd is easy. A working setup consists of three files:

timEd.exe (yes!)
timEd.hlp (the helpfile)
timEd.cfg (the configuration file)

A sample timEd.cfg is included, edit it to suit your needs. The keywords that can be used in the configuration file are explained below. Apart from that, the sample .cfg file is heavily commented.

In timEd, you can press F1 in most places to get help ('context sensitive', as we like to call it :-)

TimEd looks for the configuration and help files in the current directory (can be changed with the -C command line parameter).

```
timEd.cfg
```

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:

```
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
```

ADDRESS

=====

Use this keyword to let timEd know what your address is. You can enter up to 25 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
```

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"
```

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
```

```
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
```

```
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

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.

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!

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.

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.

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.

AKAmatching.
=====

Format: AKAmatching < Yes | No >

This option is interesting if you have more than 1 address. If you set this 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 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.

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, for some reason, 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.

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] "

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.

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?).

Example:

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

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..

```
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).

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.

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
```

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\
```


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

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.

Example: Printer LPT2

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

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

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!

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.

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

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.

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'.

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_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 are the text you so carefully edited :-)

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.

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
```

```
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).

```
MACRO  
=====
```

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 were 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!).

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.next.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.

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 Plum'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

SQUISHCFG

=====

TimEd can read the Squish.cfg (current version: 1.10) for you, and get all EchoArea lines from that configuration. Note that Netmail, bad_msgs and dupes are not read; you will have to define those manually, because you might not want them (dupes, bad_msgs) or usually want to specify special attributes for that area (netmail, local).

-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

FASTECHOCFG

=====

TimEd can read the Fastecho.cfg for you (current version: 1.30), and get all echomail areas from that configuration. Note that Netmail, bad_msgs and dupes are not read; you will have to define those manually, because you might not want them (dupes, bad_msgs) or usually want to specify special attributes for that area (netmail, local).

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

IMAILCFG
=====

TimEd can read the Imail config files for you (current version: 1.40), and get all echomail areas from that configuration. Note that Netmail, bad_msgs and dupes are not read; you will have to define those manually, because you might not want them (dupes, bad_msgs) or usually want to specify special attributes for that area (netmail, local).

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

GECHOCFG
=====

! Note: older versions of GEcho (older then 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; you will have to define those manually, because you might not want them (dupes, bad_msgs) or usually want to specify special attributes for that area (netmail, local).

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

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; you will have to define those manually, because you might not want them (dupes, bad_msgs) or usually want to specify special attributes for that area (netmail, local).

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.xml

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; you will have to define those manually, because you might not want them (dupes, bad_msgs) or usually want to specify special attributes for that area (netmail, local).

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

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
```

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)
<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;
```

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

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 netmail" wlnk_net c:\fd\wlink -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 Delmare" POINTS.133 c:\squish\delmare -P2:500/133.999
```

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

```
EchoArea "Artware testers echo" 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.

Well, if you changed `timed.cfg` to suit your needs, you can fire `timEd` up to see if it'll work.

At startup, `timEd` shows an intro-screen where you can see what config files it is currently reading (`timed.cfg`, `squish.cfg` and `areas.bbs`).

When the configuration files are parsed, `timEd` will take you to the area selection screen. You should see a list of all areas you defined. The first area will be 'highlighted'.

If you want to check the intro-screen, you can specify the `-p` command line parameter. `TimEd` will then wait for you to press a key, before going to the Area Selection Screen (like: "`timed.exe -p`".)

The Area Selection Screen.

=====

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.

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.

F1 : Show a helpscreen with available functions.

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.

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

Pressing the space bar behaves like pressing <page down>, unless you are already at the end of the message. In that case, timEd will jump to the next message in that area. So only pressing <space> will show

you all the text of all available messages in an area!

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 <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:

▶ 25 30

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

You press ctrl-right (to get a pop-up windows, 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!

Entering Messages.

In order to generate a message, you must use an external editor. TimEd will use the text generated by this editor to create a message.

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 postion 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 40 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 screwed 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, Today=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.

With this knowledge, we can look at the commands that will let you enter a message, but before that we'll look at the string editing functions.

String editing in timEd =====

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

Selecting user name and address. =====

While you are in an area, there are two functions that you might want to use:

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

You can also change name and or address while editing a header. By default, timEd will put you in the TO: field when entering a message. When you press 'cursor up', you will go to the TO: fields (the TO: name and (if applicable) the TO: address). You can now freely edit these fields, or press TAB to get a list of the names/addresses (defined in timEd.cfg) to select from.

Now the other functions to use while you are reading a message:

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 :-)

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 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 saving the message.

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>.

F1 : Help

This shows a short help-screen.

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.

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 (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 four other choices: Pack, Re-Index, Fix and Inspect. This will call SQpack, SQreidx, SQfix and SQinfo respectively (with the message area as parameter). See to it that these programs are in your path. The programs are *not* called with the extension .EXE, so you *could* place a file called (for example) sqpack.bat in your path instead, and call SQpack/386/DPMI instead :)

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\sqdfile), 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"
```

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 attributes (maybe add crash or something), 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! :-)

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.

<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).

TimEd's Find feature.
=====

ALT-F, F : Find message

This will show a small pop-up screen, where you can set the criteria for the search. It looks like this:

Areas: _____ <input type="checkbox"/> Current Area only <input type="checkbox"/> All Areas	Search for: _____ in: _____
Messages: _____ <input type="checkbox"/> From Lastread <input type="checkbox"/> All Messages	

Press F1 for help

Give up to 10 strings to search for. This will be an 'OR' search, so if *any* of the strings is found in the indicated location(s) [F)rom field, T)o field, S)ubject, B)ody] the message will be shown. You can give more than one location, so for example..

intel SB

.. is legal (search for 'intel' in Subject and Body).

The search is not case-sensitive.

Searching will start when CTRL-ENTER is pressed. ALT-R clears all strings entered.

{+} ALT-S will save the search strings that are currently defined in that area. As soon as you call the search fuction IN THAT AREA again, these search string will pop up again! The information will be added to the info files in the area (*.MSG: timEd.dat, Squish: *.SQT, JAM: *.JTI, Hudson: MSG##.TIM - see section on 'hellostrings' for more info).

Use the <ENTER> and TAB keys to move around.

When a match is found, the message will be shown, and if any matches are found in the message body, the lines with the match will be 'highlighted', using the colour specified by 'color_msgspecial' in

timEd.cfg (note: This is only functional in REGISTERED versions of timEd!).

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>.

Creating netmail and file attaches.

=====

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. 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.

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.

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.

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.

TimEd's 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.

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

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- : 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!

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>).

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.

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.

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:

When timEd is idle, it is (only) waiting for a keypress, it does nothing else (like maintaining a clock) in the meantime. Since it gets this keypress through a very standard DOS call, DESQview will give timEd no processor time at all while timEd is waiting.

OS/2 (DOS box) didn't really like it, strangely enough, so whenever OS/2 is detected (DOS version > 9), timEd will give up timeslices to OS/2. This works very well, but I would rather use the OS/2 version :)

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.

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\
```

And -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
```

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.

Some things you should know about timEd:

- TimEd doesn't really like large messages. Messages larger than 60 Kb may cause very strange behaviour (anything is possible).
- With the OS/2 version, spawning E.EXE or EPM.EXE directly from timEd/2 didn't work. Using a batchfile (with "E.EXE %1" as the only contents) worked OK...
- TimEd cannot handle Squish indexes larger than 64 Kb. This corresponds with +/- 5200 message.
- TimEd's list mode cannot handle more than +/- 16000 messages. This problem could present itself in JAM areas (that, apart from the list mode, can contain as many messages as you like: timEd will handle it).

TimEd is written using Borland C++ 3.1 (DOS version) and Borland C++ for OS/2 1.00 (the OS/2 version), and makes extensive use of Scott Dudley's MSGAPI (thanks Scott!). It also uses 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). Not the PC that is also running the BBS (I still prefer DOS to run my BBS), but another one (talking about investments!).

If someone is really enthusiastic about this program: I would very much like to use a Watcom compiler, please buy me one! :-)

TimEd source code

Not available.

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
V32bis/V42bis

FidoNet: 2:281/527

Internet: art@beard.nest.nl

Or just send me a letter, or postcard:

Gerard van Essen
Leharstraat 67
2551 LB Den Haag
The Netherlands, Europe

<End of beta documentation>