THOR

An Amiga Offline Reader
Release 2.5
Copyright © 1993-1998 Ultima Thule Software

Petter Nilsen
Eivind Nordseth
Kjell Irgens
Troels Walsted-Hansen
Borge Nost
Marius Mortensen

Copyright © 1993-1998 Ultima Thule Software, All Rights Reserved.

Published by Ultima Thule Software.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies and that no information is removed or added without Ultima Thule Software's permission.

Typeset in TEX.

General information

1 About

THOR is developed for saving online time on Bulletin Board Systems and the Internet, by allowing the user to build a database of messages that can be manipulated offline. And, of course, for the simple pleasure of creating the best offline reader on the Amiga

The team behind THOR is

Petter Nilsen, Eivind Nordseth, Kjell Irgens, Magne stlyngen, Troels Walsted Hansen and Brge Nst.

Petter is the team coordinator and creator of the THOR main program, including look and feel. He has also made ConfigTHOR, ConnectTHOR, and various other programs. He is also the maintainer of utgui.library and bbsread.library starting from 2.4 of THOR.

Eivind is responsible for the database functionality in THOR. He has also written the NComm scripts, the ABBS module, and part of the UUCP module.

Kjell has made the UUCP, QWK, SOUP, TCP (GetTCP and SendTCP), BlueWave, Hippo and Omen modules for THOR. He has also done the major part of THOR's mime compliance.

Magne has done much of the work in making THOR font sensitive and rescalable. He has made the utgui-library, which contains THOR's layout-engine and custom gadgets. He has also made the Fido modules, and done a considerable amount of work on the utnet.library.

Marius Mortensen created the internal editor which have lately been maintained by Brge Nst.

If you want to reach us with comments and/or bug reports, contact us in either of the following ways:

Internet:

EMail: `thor@thule.no` (THOR Development Team)
WWW: `http://www.thule.no/thor/`
News: `alt.sys.amiga.thor`

Snail-Mail:

Ultima Thule Software
Attn. Petter Nilsen
Mellomveien 138
N-9006 TROMSOE
NORWAY

Disclaimers

We have made every effort possible to ensure that THOR will function as described, but no warranty of any kind, expressed or implied, is made by us. We will not be liable for any direct or indirect damages to either machinal equipment or mental health arising from a failure of the program to operate in the manner desired by the user. In no event will we be liable for any loss of profits or savings.

AMIGAGUIDE SOFTWARE IS PROVIDED "AS-IS" AND SUBJECT TO CHANGE; NO WARRANTIES ARE MADE. ALL USE IS AT YOUR OWN RISK. NO LIABILITY OR RESPONSIBILITY IS ASSUMED.

INSTALLER SOFTWARE IS PROVIDED "AS-IS" AND SUBJECT TO CHANGE; NO WARRANTIES ARE MADE. ALL USE IS AT YOUR OWN RISK. NO LIABILITY OR RESPONSIBILITY IS ASSUMED.

AmigaGuide, AmigaGuide.info, amigaguide.library, WDisplay, WDisplay.info (c) Copyright 1991-93 Commodore-Amiga, Inc. All Rights Reserved. Reproduced and distributed under license from Commodore.

Installer and Installer project icon (c) Copyright 1991-93 Commodore-Amiga, Inc. All Rights Reserved. Reproduced and distributed under license from Commodore.

ReqTools is Copyright (c) Nico Franois.

2 Registration

This program is Shareware, and you must register it if you use it on a regular basis. The evaluation period lasts for 30 days. After this period, you must either register THOR, or stop using it. See details below on how to send money, and what the actual price is from your country.

When you have registered THOR you will receive a personal key file that will work with future versions of THOR. You will also get free support, and last, but not least, a clear conscience.

Please see the included registration files for more info.

The First Virtual registration scheme can be used whereever you are, as long as you have a valid MasterCard or VISA credit card. Please see `http://www.fv.com/` for details.

If you already have First Virtual Pin, you may register Thor using out online First Virtual registration service at `http://www.thule.no/cgi-bin/member`.

Features in THOR

What is so special about THOR?

Several things. First, it is as far as we know the only general offline reader for the Amiga with true 'history'. This means that all messages will be added to a database, and may be recalled later. Threads may be followed forwards and backwards.

Furthermore, THOR supports more different message formats than any other offline reader on *ANY* platform, and new ones are easily added because THOR is a very open system. At this point THOR supports [ABBS/MBBS], page 29, [QWK], page 29, [BlueWave], page 30, [SOUP], page 30, [UUCP], page 33, [FIDO], page 31, [TCP], page 35, [Omen], page 30 and [Hippo], page 29 (BBBS) systems.

THOR also has a name database that contains data on people who have written messages on the system you are using. This makes it easy to enter a message to someone, just enter a few letters from this persons name and a window will come up containing suggestions based on the letters you wrote.

THOR uses an advanced buffering system to be able to get best possible performance on disk operations. Copy-back buffering is used when messages are added to the database by the different parser programs, and write-through buffering is used elsewhere. The number and size of buffers can be set in the Global Configuration in ConfigTHOR.

THOR sports a complete Style Guide compliant 2.x-look GUI with extended features if run under OS 3.0 or higher. Since gadtools.library from Commodore doesn't support color in listviews easily, we have made our own custom listview to be able to have color and ANSI in the listview. THOR is also completely font-sensitive and most windows can be rescaled and their position saved. Selection of screenmode, fonts to use, palette etc. is also supported. Since many of the gadgets are custom made by us, many of the features available previously only to OS 3.x users, are also available for OS 2.04 users. THOR uses pen sharing under OS 3.0 or higher, and a separate background picture in the windows and on the screen is supported. This feature uses the datatypes.library and the pen sharing available in OS 3.0 and higher.

THOR fully supports the 24-bit picture.datatype.

THOR has online help using the AmigaGuide system from Commodore, so whenever you want help, pressing the HELP key will bring up help on the window or the menu under the pointer.

THOR has an advanced built-in editor called the Full Screen Editor (FSE) This editor will satisfy most users, but you can of course use an external editor if you like. See Chapter 9 [THOR Full Screen Editor], page 43.

THOR has a flexible system for purging conferences based on number of days and/or maximum number of messages. XPK packing of the message database is also supported directly with a high degree of flexibility.

THOR has a high-speed search function which lets you search for a given key word, with several different search options.

THOR has an internal filedatabase that can be used for keeping e.g. a full list of files on Aminet or from the systems you call.

THOR has an extensive Kill/Emphasize database that will let you kill or put an emphasize on messages based on any criteria in the message data, with wildcards.

THOR has extensive Arexx support and includes a lot of useful arexx scripts. These includes uuencode, crosspostings, mailinglists, chess(!) etc.

THOR has locale support, and supports a wide range of international character sets, like ISO, IBM, NO7, SF7, DE7, etc. If you wish to translate THOR to a language not already supported, please contact us.

THOR has 13 ways of marking a message or a range of messages, e.g. mark messages to you as unread, mark all messages with a certain subject as read or mark messages from a user as read.

THOR has extensive Chapter 12 [MIME support], page 55 for the Internet news and mail types, including base64 file attachment and quoted-printable. Received MIME pictures can even be viewing inline in the listview!

3 Installation

A hard disk is required to use THOR.

Installation should be done with the installation script. Just click on the Installation icon and follow the instructions.

If you only intend to use THOR with the [TCP], page 35 system type on the Internet to get and send mail and news using NNTP, POP3 and SMTP, you may skip to the [TCP], page 35 section now.

THOR supports the use of scripts in communication programs. It comes with Ncomm scripts for ABBS, MBBS and BBBS and a few other specialized Ncomm scripts. We recomend that you use these scripts when possible.

The script file will be named <mail-packet name>.script (for example thule.script) and it will reside in the <Your THOR path>/Scripts directory.

It is possible to make your own scripts for other system types and communication programs (Check out Chapter 8 [Scripts and communication], page 42).

Starting THOR for the first time

When you first start THOR, you will be asked to set up your Global Configuration.

There will not be any available systems in the [Startup Window], page 9 at this time. First, you must tell THOR some details about the systems you wish to add. Choose the Systems Configuration subitem from the Configuration menu. This brings you to the Systems Configuration Window. Note that if you complete the installation procedure during installation and set up a system there, this is not needed.

Configuration

The configuration program for THOR is called ConfigTHOR. A separate documentation for this program is available in ConfigTHOR.guide.

THOR Tooltypes and arguments

When starting THOR, some arguments may be given to THOR, either on the command line, or in the icon tooltypes:

`PUBSCREEN <name>`
: This is the name of the public screen you want THOR to open on. Please note that this name is case-sensitive up until V40 of the OS. If the public screen does not exist, a new screen by this name is opened.

`FORCETOFRONT`
: If THOR is already running, its screen will come to front. Without this option, a new THOR will be started.

`SYSTEM <name>`
: THOR will enter this system directly without opening the Startup Window. Any waiting message packets will be parsed in as default with this option.

`NOPARSEGRAB`
: Used in conjunction with the SYSTEM option THOR will not parse message packets from the configured the download directory.

`ICON_X`
: This is the X coordinate that THOR will use for the icon when iconified.

`ICON_Y`
: This is the Y coordinate that THOR will use for the icon when iconified.

`SCREENPICTURE`
: This is the name of the picture you want as background picture on THOR's own screen. Can be in any format supported with a datatype. This argument requires that you have OS 3.0 or higher installed and will override any screen picture settings in the configuration.

`WINDOWPICTURE`
: This is the name of the picture you want as background picture in the windows in THOR. Can be in any format supported with a datatype. This argument requires that you have OS 3.0 or higher installed and will override any window picture settings in the configuration.

4 Windows in Thor

Visual configuration window

This window is used to configure message appearance. The following options are available in this window:

`Color Quote`
: Toggles color quoting. If color quoting is on (the gadget has a checkmark), quoted lines appear in a different color. This option is also used by the editor.

`Double Quote`
: Toggles double color quoting. If double color quoting is on and the *Color Quote* option is ON (the gadgets has a checkmark), two different colors are used for quoting:

```
>>This is a line in one color
>This is a line in another color

This option is also be used by the editor.
```

`ANSI Support`
: Toggles ANSI Support. If on, all messages that include ANSI sequences are shown in full ANSI color.

Note: To show ansi bulletins and messages with their true ANSI colors, you must be running on a 16 colors (or more) screen under OS 3.0 or higher.

`Style Tags`
: Toggles Style Tags support when reading messages. Style Tags is a way to "mark" single words and is only available if *ANSI Support* is on. Here are some examples:

```
*GREAT*       -  The word "GREAT" is shown in Bold

_GREAT_       -  The word "GREAT" is shown in Underlined

/GREAT/       -  The word "GREAT" is shown in Italic

#GREAT#       -  The word "GREAT" is shown in Complement.
```

Style tags may be combined; for example:

```
*_GREAT_*     -  The word "GREAT" is shown in Bold and
                 Underlined.
```

But this won't work:

```
*_GREAT*_
```

Starting from THOR 2.3, several words can also be styled, like this:

```
/This is an italic text/
_This is an underscored text_
```

`Show Pictures`

`Manually`
: Attached pictures can only be viewed by manually selecting it using the icon in the listview.

`Auto Internal`
: Attached pictures will be shown inline automatically.

`Auto External`
: Attached pictures will be shown automatically using the external viewer defined for pictures. See Chapter 12 [MIME support], page 55.

`No Links`
: Selecting this option will make THOR not search for any http:, ftp: etc. links in the message text. If speed is your preference, and you're not interested in the power clickable links gives you, you should select this option to disable all link detection.

`Link Type`

`Button`
: Links will appear as buttons in the message listview with this option.

`3D Text`
: Links will appear as white text with a black dropshadow in the message listview with this option.

`Underscore`
: Links will appear in a in a separate colour with underscored text, as seem in WWW browsers.

`HTTP Program`
: This is the program or ARexx script that will be started when someone clicks on a http: link. The only argument given to the program/script will be the url clicked on. IBrowse, Voyager and AWeb has been tested to work directly. Default entry is the http.thor ARexx script.

`HTTP Mode`

`Run ARexx script`
: Execute the ARexx script in HTTP Program passing in the URL as the first argument.

`Run program`
: Run the program in HTTP Program passing in the URL as the first argument.

`FTP Program`
: This is the program that will be started when someone clicks on a ftp: link. The only argument given to the program/script will be the url clicked on. AmiTCP has been tested to work directly. Note that even if you use the internal ftp in THOR, you should configure a ftp program that will be used on ftp:// URLs that only contain a directory and not a filename. FTP can be configured to use external (the program you've defined here) or the internal ftp support.

Packing and purging window

The messages in the database will remain there until you decide to pack and purge conferences. Purge in this context mean to *delete* old messages from disk, and there is no way to get them back.

Messages marked as deleted with the "Delete Msg On/Off" menu option will also be physically deleted from disk.

If a Xpk-method is defined in the configuration, remaining messages will be compressed with the defined Xpk-method.

THOR can purge a conference based on either global values or separate values for each conference. The values for each conference may be set in the Conference Configuration (in the ConfigTHOR program) and the global values are set in the Global Configuration.

There are two basic methods for purging conferences, one is based on number of messages to keep in each conference, the other is based on the number of days to keep messages. Both of these can be used at the same time.

If you set Msgs to Keep to '100' and Days to Keep to '14', all messages older than 14 days will be deleted, and no more that 100 messages are kept.

One or both of these values may be "ignored", which means messages are not deleted.

All conferences will use the global setting as default. Therefore, it is enough just setting the global values and use these on all conferences. If there is a conference or two that might interest you more than the rest, you typically set higher values for these.

From of release 2.3 of THOR, a new window for this purpose has been added to let you select which conferences to pack/purge in a more controllable fashion.

This window offers the following gadgets:

`List of conferences`
: This is a list of conferences on the system with unread and total messages shown to the left of the conference name.

`Pack/purge in`
: This can be set to "All confs" (default) or "Selected confs". The last selection will only pack/purge the selected conferences in the list.

`Start`
: Will start the pack/purge on all or selected conferences.

The listview containing the conferences can be switch to a list of attachments that were attached to the messages that was purged. If the pack/purge deleted messages that contained attachments, the window will switch to this attachment listview when you are done packing/purging. You may then delete, copy, etc. the attachments.

Startup window

This is where you select a system to read messages from. It should be fairly self-explanatory as the only functions are 'Rescan' and 'Open' along with a pick list of systems.

'Rescan' looks for new mail packets in your download directory while 'Open' enters the selected system.

The 'Systems' list needs a little explaining. Displayed here are the names of all bulletin boards / Internet systems you have configured in Systems Configuration in ConfigTHOR. You may select any of them to open for reading.

You can select a system in one of these ways:

- Double-clicking with the mouse on a system opens this for reading and un-archives any message packets that might be waiting for this system (Same as pressing Open.)

- The cursor keys can be used up or down to select a system.

- Pressing Return/Enter on the keyboard or 'o' for 'Open' opens the currently selected system.

- The 'Open' gadget opens the currently selected system.

This list also has some "flags" to the left of the system name which are used to give you information:

* - This means that you have 1 or more mail packets waiting for this system in the download directory.

U - This means that there are unread messages on this system.

E - This means that you have events outstanding on this system.

+ - This means that there are unread messages to you on this system.

NOTE: Holding down one of the Shift-keys on the keyboard while entering a system, will make THOR skip the parsing of any message packets that might be waiting in the download directory.

Main reader window

This is the reader window, and you don't need any other windows open to use THOR. The Main Window contains all information on the current message (sender, receiver, date, subject, etc.) along with the message text. See [Message header], page 53.

The Main Window also contains a status line where conference, internal (THOR) message number, system name and various statistics are shown. It is also used for displaying various messages.

The size and position of all these windows are saved along with the visual options. See [Visual Options Window], page 6.

The buttons in the Main Window are explained in detail under Chapter 11 [Message reading], page 49.

Newfiles window

Many systems send you a list of new files since your last logon. This window is used to display that list. If the system type for this system supports downloading of files, clicking on a file in the Newfiles Window can be used as an interactive way to get files from the system. Scrolling horizontally and vertically in the list is supported with the cursor keys in combination with the Shift key.

This window's size and position is saved when you save the visual configuration. If the window is open when you save, it is opened automatically in your next session.

Function dock window

Not all System Types will support every function. Gadgets not supported by the System Type will not be present in the dock.

Getting a list of conferences/newsgrops

Choosing this function will request the conference list the next time you call the system. This is required for THOR to update the conference flags correctly. This function is not available on all system types.

Download a file

This brings up a requester asking for a filename for a file that will be obtained from the remote system on the next call. This function is not available on all system types. On TCP system, you may type in a full URL for the file to have it downloaded for you.

Example: `ftp://ftp.thule.no/pub/thor/thor25upd.lha`

Upload a file

This window is used for uploading a file to the current system. The following options are available:

`System Filename:`
: This is the name the file shall appear as on the system. This name can differ from the file name on disk. The "File" gadget brings up a requester with your configured upload directory as default. It is not possible to enter a name here without having selected a file on disk first.

`Description:`
: You can enter a short file description here.

`PrivateTo/Private:`
: This is used if you want to make the upload private to a person.

`Conference/Private:`
: If you fill in this field, the uploaded file will only be available for members of this conference. You cannot both make an upload private to a person and to a conference. No checking is done to see if files actually CAN be uploaded private to the conference you enter here, so be SURE that the conference accepts files private to it!

`Detailed Description:`
: If you choose this flag, the editor is brought up when you choose "Ok" and you can enter a detailed description for the file.

`Download Notification:`
: Lets you have a notification when anyone downloads the file from the system. This function is not available on all system types.

The upload function is not available on all system types. If upload is ghosted you will have to do this manually.

Resign/unsubscribe a conference/newsgroup

Choosing this function brings up a requester with the current conference's name as default. You can enter any valid conference name here and you will be resigned from this conference next time you call the system.

Join/Subscribe newsgroup

You may select conferences from the list shown here, and you will be made a member of this conference, provided that this is not a restricted area on the system. This window will also let you use wildcard search (with automatically selection) for conference names. You can use multiselection to select more than one conference.

Send Userinfo

On system Types supporting it, this function will let you enter information about yourself for other users to examine.

Catchup unread messages

This function marks *all* unread messages in the current conference as read (will not effect messages with the SuperUnread flag set).

Show Unread/All/Search result

THOR operates in three basic modes, "Unread", "All" and "Search". The Unread mode is used for reading only new messages, the All mode lets you read all messages, while the Search mode is used for browsing messages found in a previous message search.

Some aspects of the message reading behaves different in these three modes. Messages which are unread by you will not be marked read if in All mode, the list of conferences in the [Conference Window], page 14 shows all conferences when in All mode and you are not allowed to choose between only viewing the conferences with unread messages in and all conferences.

The [Messagelist Window], page 13 shows only the Unread messages when in Unread mode, while it shows all messages in All mode. In Search mode, it will show all conferences with messages found in a previous message search.

If you follow a thread backwards in All mode, the nodes in the [Messagelist Window], page 13 will be updated and the current shown message will be highlighted to reflect this. This works differently in Unread mode, since only unread messages are shown. If you follow a thread backwards in Unread mode, the list of messages in the Message list Window will not be updated and will still highlight the message you started threading on. For Unread mode, this is useful since choosing Next msg will bring you straight back to the message after the one you started the threading on.

Grouping of messages

Messages messages may be sorted in several ways useful for reading. Starting from THOR 2.5, not just unread messages may be grouped, but also messages in All and Search mode. This gadget is used to change the mode of reading to one of these:

`Reference`
: This mode will sort the messages and group the different threads together.

`Mine First`
: This mode will sort the messages so that all messages to you will come first. Other messages will be sorted in reference order.

`All First`
: This mode will show messages addressed to ALL first. Other messages will be sorted in reference order.

`MsgNr`
: Will show the messages in the order they were added to the database.

`Subject`
: Groups the messages by subject. Very useful for systems with a lot of messages, such as UseNet News.

`Grouped`
: Groups the messages by subject, keeping each similar subject folded. Very useful for systems with a lot of messages, such as UseNet News. This group method is only useful with the message list window open.

`Subject Alpha`
: Groups the messages by subject alphabetically.

`Grouped Alpha`
: Groups the messages by subject alphabetically, keeping each similar subject folded. This group method is only useful with the message list window open.

`Author`
: Groups the messages by author name alphabetically.

`Grouped Author`
: Groups the messages by author name alphabetically, keeping each similar author name folded. This group method is only useful with the message list window open.

This setting can be saved on a system or global basis using Visual Options from the Configuration menu.

Open the events window

Brings up a window containing all the events you have pending on the system. From here, you can edit, delete, freeze and activate events. See [Event Window], page 21.

Message overview/list window

This is an overview window which will contain the vital information about a message, like the sender, receiver and subject of the message. The current message will be highlighted with a bar. When you read messages, the list in this window will scroll accordingly.

If a conference is set up by the packet parser to only contain messages addressed to ALL, this window will not contain a receiver (E.g. Usenet News). The window will change on-the-fly to reflect the setting for each conference.

You can use the mouse and shift key to select multiple messages to perform some operation on. Currently, you can save, delete, set/remove the superunread flags, set/remove the keep flag, move and print the selected messages.

The small gadget at the top right, above the scrollbar on the right, can be used to selected/deselect all messages in the listview.

Note: If you have many messages in a conferences, this might take some time. If you are a 68000 user, it might be best to leave this window closed in these cases.

To the very left if this window the following flags might show up:
K - The message is marked with the KEEP flag.
D - The message is deleted from the message database.
U - The message is unread.
I - The message is either marked as Important or as Urgent
S - The message is marked as SuperUnread.
* - The message is to you.

Conference overview window

This window will contain the conferences available on the current system. A cycle gadget at the top of window that will let you show all conferences or only those with unread messages in. The status of this gadget is saved by the [Visual Options Window], page 6. If you are reading messages in ALL mode , this gadget will be in its ALL state and cannot be changed. See [Show], page 12.

The current conference will be highlighted with a "bar" in the window.

To the right of the conference names are several flags which will signal what type of conference this is. These flags are not set unless you use BlueWave format or have collected the conference list. See [Get ConfList], page 10.

The flags have the following meaning:

```
+ - Unread messages to you present in this conference
* - Member of this conference
C - Conference is compulsory
M - Conference is a mail conference
U - Userinfo
R - Read-Only conference
F - Fileinfo
P - Private allowed
```

The number of messages unread in each conference will be shown before the conferences name when in Unread mode, and the total number of messages will be shown when in All mode. See [Show], page 12, for more info.

When you are reading in Search mode, the number of messages will remain constant on the number of messages found in the last message search in this conference.

A new conference can be selected for reading by either using the mouse to click on the conference, or by using the cursor keys to scroll the list and select a conference with Return/Enter.

File database window

Each system in THOR can have its own file database. This is the window that is used to perform operations on this file database.

This file database can be used to maintain a list of files from a BBS you call, or a list of file on Aminet, etc.

The operation of this window might seem confusing at first, but it's fairly straight-forward. Several gadgets are available in this window:

`Areas`

When this gadget is pressed in, the list in the window will contain the different file areas available in the database gadget. Several areas can be selected for some operation by dragselection with the mouse.

`Files`

This is a "virtual" area where files can be sorted etc. If you have any file areas selected and then press this gadget, all files in the selected areas will be added to this list. If you select any files here, and then press the "Hold" gadget, the selected files will be moved to the Hold virtual area. Files can be download, deleted etc. from here. When leaving this virtual area, the files will still remain.

`Hold`

This virtual area functions the same as the "Files" area, except that any files found in a search will end up here.

`Date`

This is a date that will be used when searching for files. Note: If the date is todays date, it will not be used when searching, it will only be used if it is anything else than todays date.

If this date is used when searching (which means it's something else than todays date), only files from this date will be found in a search.

By setting a date to e.g. 10 days ago, and searching for "*", all files newer than 10 days will be added to "Hold".

`Maintenance`

When files are deleted from the database, they still take space on disk. By using this function, these files will be physically deleted from disk and won't take any space any longer. As an added bonus, the filedatabase will also be optimized and the hash tables will be expanded if the database is large.

`Download`

By using this function, you can add download events for the files you have selected. This is only available on system types supporting download.

`Search`

This function is used to search in the filedatabase. Two types of searching is available:

1. Exact filename search, which means that the search will find any file matching the filename given *exactly* . Wildcards are supported, so a search for "ABBS#?" will find all files starting with "ABBS" in the name.

2. Keyword search, which means that the search will include both filenames and descriptions. Standard AmigaDOS wildcards can be used, and all found files will end up in the "Hold" area.

The search can be restricted by selecting the File Areas to search in before selecting this function.

`Sort`

This function will, if selected while Areas is the active view, sort the areas in the database either on disk or only in the window. If "Files" or "Hold" is active, the files will be sorted in the current view. If sorting files, you can also selected whether to sort after filename, number of downloads, date, or description.

`Remove`

This function will remove from the window the selected areas of files. It will not have any effect on the organization of the areas/files on disk.

`Delete`

This function will delete the selected areas/files from disk. Use of "Maintenance" might be useful if you have deleted any areas of many files.

`Clear/Mark`

If any areas/files are selected, this function will de-select all selected items.

If no areas/files are selected, this function will instead select all items in the current view.

`Save`

This function will let you save to disk or clipboard the all/selected areas/files in the current view. The list saved to disk/clipboard will look exactly like the list shown in the current view.

`Invert`

If any areas/files is selected, this function will deselect these any instead select those previously not selected.

Several ARexx-scripts are available the uses the filedatabase for various operations, here are a few of them:

In the "rexx/" directory:

AddFileList.br - Adds a ABBS filelist to the database. No checking for already existing files is performed.
AddAmiNetFileList.br - Adds a Aminet INDEX/RECENT file list to the database. No checking for already existing files is performed.
ClearFileDataBase.br - Clears the file database for the given system.
FindFile.fse - Lets you search in the database from the internal Full Screen Editor in THOR and include the found files at the current cursor position.

Several example scripts using the file database is also available in the "rexx/examples/" directory.

The position and size of this window is saved with the rest of the settings in the [Visual Options Window], page 6.

User database window

The user database window consists of an overview of the users in the database, and some options to add and edit user records.

This window consists of several gadgets:

`Users in database`
: This is a list of all the users on the user database, sorted in the order they were added to the database. To the far left is the user name, the next field is for any alias that might be configured for this user. Next comes the date this entry was added to the database, and to the far right is any comment that might be associated with a user. Several users can be selected for deletion with multiselection using the mouse and the shift key. Double-clicking on a user will open the enter message window, ready to enter a message to that person.

`Sort`
: This function will sort the user entries in the window alphabetically, but the result will *not* be updated on disk.

`Delete`
: This function will delete the selected user(s) from the user database. That is, the users will only be *marked* as deleted, but won't physically be deleted from disk until the "Maintenance" function is used.

`Save`
: This function will update the currently selected user with the information contained in the various string gadgets in the window.

`New`
: This function will let you add a new user to the database. When you have finished entering the information you want, select "Save" and the user will be added to the user database.

`Maintenance`
: This function will optimize and remove out deleted entries from the user database.

`Name`
: This is the user name of the currently selected user. This field will be used for loopup when entering messages.

`Alias`
: This field contains the alias you want the user to be known under. By using this field wisely, e.g. all you close personal friends could be given the same "friend" alias and by entering "friend" in the username field when entering a message, you will be given the choice to send the message to all users with "friend" as alias.

`Address`
: This entry usually contain the email address (or the equivalent depending of the system type). This entry will also be used for loopup when entering a message, although the user name got a higher priority.

`Comment`
: This field is for adding any comment you might want to an user entry.

A typical user entry on Internet might look something like this:

```
   Name: Petter Nilsen
  Alias: thor-guy
Address: pettern@icenet.no
Comment: This guy is one of the THOR authors
```

See the Arexx-script Adduser.thor located in rexx/ for an example of using the user database from ARexx.

The position and size of this window is saved with the rest of the settings in the [Visual Options Window], page 6.

Kill/emphasize window

This window contains one of the most powerful features added for THOR 2.0, namely a way to sort out unwanted and special messages automatically. A "kill" option has been fairly common in Usenet news-readers for a while now. With the introduction of THOR 2.0, we expanded this concept further.

A "kill" entry is basically a description on what messages you do *not* want to read when reading messages. If e.g. you do not want to see any messages from the user "Marc Barrett", you simply add a kill on the user name "Marc Barrett", and all messages matching this "kill" will be marked as *read* automatically when imported into THOR's database, so you will not see the message when reading normally. 2.5 takes this concept further by having a new *real* kill mode. This new mode will drop all messages matching the kill, and they will never be imported into the database.

Likewise, an "emphasize" entry will notify you about any matches found for this "emphasize" entry. If e.g. you want to be notified specially about messages from "Dave Haynie", you simply instruct THOR to mark all messages matching his name as "Important" or "Urgent". A *Haze* level can also be used when "emphasizing" messages (more about this later).

The Kill/Emphasize database consists of two windows. The first window is for listing of the current active kills/emphs and contains the following gadgets:

`Active Kills/Emphasizes`
: This will contain all active kills/emphasizes in the database. By using the mouse of the cursor keys, you can select a kill/emphasize for editing or deletion.

`New`
: Selecting this function will open the kill/emphs edit window with all fields unselected. You can now enter the information you want this kill/emph to contain.

`Edit`
: This kill/emph edit window will be opened with the information from the currently selected kill/emphasize.

`Delete`
: This function will delete the currently selected kill/emphasize.

`Maintenance`
: This function will remove deleted entries in the beginning of the list of kills/emphasizes.

When selecting either *New* or *Edit* from the window listing the current kills/emphasizes, a new window will open with several gadgets that allows to you be very specific about the information you want to trigger the kill/emphasize on message:

`From-Name`
: This is the name of the sender of any message THOR is to trigger the kill/emphasize on. When the checkmark is not set, this field is not used. Standard AmigaDOS wildcards are allowed.

`From-Address`
: This is the address of the sender of any message THOR is to trigger the kill/emphasize on. When the checkmark is not set, this field is not used. Standard AmigaDOS wildcards are allowed.

`To-Name`
: This is the name of the receiver of any message THOR is to trigger the kill/emphasize on. When the checkmark is not set, this field is not used. Standard AmigaDOS wildcards are allowed.

`To-Address`
: This is the address of the receiver of any message THOR is to trigger the kill/emphasize on. When the checkmark not is set, this field is not used. Standard AmigaDOS wildcards are allowed.

`References`
: This entry is only useful on Usenet-type of systems where messages contain a "References:" field. This field can contain reference strings that THOR is to trigger the kill/emphasize on. When the checkmark is not set, this field is not used. Standard AmigaDOS wildcards are allowed.

`Subject`
: This is the subject of any message THOR is to trigger the kill/emphasize on. When the checkmark is not set, this field is not used. Standard AmigaDOS wildcards are allowed.

`Msg Text`
: This field can contain any word combination in the main message body that THOR is to trigger the kill/emphasize on. When the checkmark is set, this field is used. Standard AmigaDOS wildcards are allowed. Note that this field must match a complete line in the message text, so if you want eg. to trigger on all messages containing "PC sucks" you need to make this one "#?PC sucks#?",

`Conferences`
: This field contain the conference this kill/emphasize will be used in. When the checkmark not is set, this field is not used. Standard AmigaDOS wildcards are allowed. Any conference can be selected from the list of conferences in THOR by pressing on the select button with the left mouse button.

`Type`
: 4 types of trigger methods are available when setting up a kill/emphasize, *Kill* , *Mark Unread* (used to be Kill), *Keep* , *Important* and *Urgent* .

Kill
By using this method, all messages matching the information given will *not* be imported into the THOR message database. This is a new method for THOR 2.5.

Mark Unread
By using this method, all messages matching the information given will be marked as *read* when imported into the THOR message database.

Keep
By using this method, all message matching the information given will be marked with the *KEEP* message flag when imported into the THOR message database. This means that this message will remain in the database until you choose to clear the *KEEP* flag manually from the message. Packing/Purging of conferences will not affect any message with the *KEEP* flag set.

Important
By using this method, all messages matching the information given will be marked with the *IMPORTANT* message flag when imported into the THOR message database. When such a message is read, THOR will DisplayBeep the screen as if the message was addressed to you. The top of the message body will also contain the words "*** This message is marked as IMPORTANT ***" in bold letters.

Urgent
By using this method, all messages matching the information given will be marked with the *URGENT* message flag when imported into the THOR message database. When such a message is read, THOR will DisplayBeep the screen as if the message was addressed to you. The top of the message body will also contain the words "*** This message is marked as URGENT ***" in bold letters.

`Haze`

This feature is named after Dave 'hazy' Haynie.

On emphasize/kills, any matching message can be given a special *Haze Level* . The various haze-levels are defined in the Global Configuration in ConfigTHOR and defines the minimum number of days the message is to remain in the messages database before it can be purged.

Tip: If you want to keep your own messages (the ones you write yourself) in the database somewhat longer, setup a *Mark Unread* trigger on your own email address, then set a haze level that'll keep the message in the database longer than other messages.

`Comment`

This is a new addition in THOR 2.5 and allows you to add a comment to each kill/emphasize you add. The comment is purely for information purposes and will not affect the operation of the trigger.

`Save`

Will save the current kill/emphasize to disk. The next time you import messages into THOR on this system, this kill/emphasize will be checked.

`Get Current Msg`
This function will fill in the available information from the current message shown in the main window (if any) in the appropriate fields in the window.

`Cancel`

Cancel the editing.

A typical emphasize might look something like this:

```
+-------------------------------------------------------------+
|  From-Name: Dave Haynie       From-Address: #?scala.com     |
|    To-name: <none>              To-Address: <none>          |
| References: <none>                Msg Text: <none>          |
|    Subject: <none>             Conferences: comp.sys.amiga.#? |
|                                                             |
|       Type: Important                Haze: Level 1          |
|    Comment: Cool hardware guy                               |
+-------------------------------------------------------------+
```

NOTE:
Use as few fields on a kill/emphasize as possible. Useless information will only make the prosessing slower and might even make you miss a kill/emphasize if you are too strict on the information supplied. Less is more.

For a kill/emphasize to match a message, all configured fields must match.

Event window

In this window all events pending on a system are displayed. You can navigate through the list by using the cursor keys or the mouse. Some functions are available:

`Edit Data:`
: Lets you edit data on the event. Most events can be edited.

`Edit Text:`
: Lets you edit the message text assosiated with most events.

`Delete:`
: This function will mark the highlighted event as deleted. You can still undelete (with activate) the event until the ScriptServer, has marked all events as done or you have uploaded the reply file successfully (with QWK/Hippo/SOUP/Fido). See [ScriptServer/CfgType], page 61.

`Freeze:`
: Delays the event (put it on ice) until the Activate function is used.

`Activate:`
: Will activate a frozen or deleted event.

The position and size of this window is saved with the rest of the settings in the [Visual Options Window], page 6.

Search window

This window is used for searching in the message database. The following options are available:

`String`

The string you want to search for. The search is not case-sensitive. Several can be defined by separating the strings with a ',' (comma). All the given strings must be found in a message if it is to be treated as a search "hit". Standard AmigaDOS wildcards are allowed.

`MsgText+Subject/Subject/None`

MsgText+Subject:
The message search will include both the message body and the subject.

Subject:
The message search will only include the subject.

None:
No text will be search for. By setting this option, and any of the fields *From user* , *To user* or *Kept* , all messages matching any of these fields will be found in the search.

`Start Msgnr`

This field is an optional message number to start the search from.

`End Msgnr`

This field is an optional message number to end the search on.

`From user`

If you enter a username here, the search will be restricted to messages from this user.

`To user`

If you enter a username here, the search will be restricted to message to this user.

`Conferences`

This field might include one or more conferences separated with a ',' (comma). The conferences to conduct the search in can be selected in the window that pops up when selecting the popup gadget to the right of this gadget. *All Confs* will override any conference(s) entered here.

`All Confs`

If this options is selected, the search will be performed through all conferences on this system. This option will override any conferences you might have entered in the "Conferences"-field.

`Kept`

The search will be restricted to messages marked with the KEEP flag.

`Search`

Starts the search using the supplied parameters.

`Cancel`

Cancels the search.

All messages found in the search can be marked as unread or you have the option of entering a special Search read mode that will function mostly like the other read modes. See [Show], page 12.

Mark window

This window is used for marking a message or a range of messages in different ways. Note: All operations that require a message number or a range of messages numbers, will need the internal message number from the message database, and not any number that the message(s) might have on the remote system. The mark operations will only apply to the current conference. The following options are available in this window, with the option's shortcut enclosed in [] brackets:

`[ENTER] Mark current message as unread`
: Will simply mark the current message as unread. The shortcut 'M' (shift-m) can be used directly from the Main window and will do the same without opening this window. See Chapter 11 [Message reading], page 49.

`[S] Set 'last message read' pointer`
: Will ask you for a message number in the current conference. All messages from (and including) the message number given will be marked as unread.

`[#] Mark message with given number as unread`
: Will mark the message with the given number as unread.

`[D] Mark messages from given date as unread`
: Will mark all messages that was written on and after the given date as unread. The format of the date is DD-MM-YY, and the operation will not be executed if the date given is in a unknown format.

`[G] Mark a group of messages as unread`
: This option will ask for a range of message, and will mark all messages within this range as unread.

`[T] Mark messages in this thread as unread`
: Will make the current thread of messages a unread. Please note that this option, as all other thread functions, requires that thread information is available on the system you are reading messages on.

`[U] Mark messages in this thread as read`
: Will make the current thread of messages a read. Please note that this option, as all other thread functions, requires that thread information is available on the system you are reading messages on.

`[B] Mark messages with this subject as read`
: Will mark unread messages in the current conference with the same subject as the current message as read.

`[O] Mark unread msgs not from or to you as read`
: This options will mark all unread messages which is not to or from you as read.

`[R] Find and mark messages to you as unread`
: Will ask for a "from" and "to" message number. All messages in the current conference within the given range that is addressed to you will be marked as unread.

`[Z] Mark unread msgs from current author as read`
: All unread messages from the current author of the current message will be marked as read. Messages addressed to you will NOT be marked as read. A temporary local "kill" without the risk of losing messages addressed to you. Use it wisely.. :-)

`[L] Mark messages from login state as unread`
: Will set the message database to the unread state it was when you entered the system in the current session. This function is useful if you want to browse through the unread messages, or you regret having used CatchUp etc.

`[C] Mark msgs from conf. login state as unread`
: Will set the message database for the current conference to the unread state it was when you entered the system in the current session. This function is useful if you want to browse through the unread messages, but still wants to read them again later for doing a reply etc.

`Cancel`
: Will abort and close the Mark window.

`Ok`
: Will start the marking of messages, with the currently selected mark option. If you use the shortcut from the keyboard for an operation, you will not have to press Ok.

5 Signature system

There are three levels of signature configuration in THOR. These are the Global, system and the conference signature. The priority is:

1) Conference signature.
2) System signature.
3) Global signature.

If a conference signature is configured, this is used when entering or answering messages. If the conference signature does not exist, the system signature will be used. If this does not exist either, the global one is used.

If no signature is configured anywhere, no signature will be added to messages.

Multi-line signatures is possible by adding "\n" (backslash n) where you want new lines.

If you want the version number of THOR inserted in the signature automatically, you can use $ver in the signature somewhere, and correct version number will replace the $ver string in the message.

Inserting $time in the signature will add the current date and time.

6 Menus

THOR has a menu strip accessable from all windows, except for the Chapter 9 [THOR Full Screen Editor], page 43 which has its own menu strip.

From the left of the screen:

`Project`

`About`
: Shows information about THOR and registration info.

`Leave System`
: Exits from the current system and returns you to the [Startup Window], page 9.

`Iconify`
: Closes the THOR screen and generates an AppIcon on the WorkBench screen. Double-clicking on this icon will open THOR again. This item is only available from the [Startup Window], page 9 and the [Main Window], page 9.

`Public Screen`
: Pops up a window where you can choose which public screen THOR should be run on.

`Print Message`
: Dumps current message in the main window to the printer.

`Save Message`
: Save current message to disk.

`Quit`
: Quits THOR.

`External`

`MLManager`
: Starts the MLManager program. MLManager is available separately from Aminet.

`ThorStat`
: Starts the [ThorStat], page 57 program. By setting the screen for this program to Thor.1 in MUIPrefs, this program can also be opened on the THOR screen.

`PackBBS`
: Runs the external shell program "packbbs" located in bin/ in a output window on the THOR screen. This program will pack all conferences on all systems based on your global and conference settings. See [Pack/Purge Window], page 8, and [PackBBS], page 59 for more info.

`ListEvents`
: This program lists all events outstanding in a output window on the THOR screen, using the program bin/listevents. See [ListEvents], page 59.

`ConnectTHOR`
: Will start the [ConnectTHOR], page 36 program for getting Internet news and mail on the screen THOR is running on.

`Configuration`

`Config`
: Starts the ConfigTHOR program in the main startup window.

`Systems Config`
: Starts the ConfigTHOR program in the Systems Configuration window.

`Conf Config`
: Starts the ConfigTHOR program in the Conference Configuration window. Not available from the startup window.

`Visual Options`
: Opens the [Visual Options Window], page 6 or brings it to front if it is already open.

`General`

`Search`
: Opens the [Search Window], page 21 which will allow you to search for specific strings based on what you select in this window. See [Search Window], page 21.

`ROT-13`
: Applies ROT-13 to the current message in the main window. ROT-13 is also know as the "Caesarian encryption" and is basically a rotation of each character 13 places up, with wrapping if the ascii value gets past the 'Z'.

`Show ScriptMsg`
: Shows the file "Scriptmsg.txt" which the script can create for various purposes, like error messages from the script.

`Delete Dups`
: Scans through the message database and deletes duplicate messages with the same message number on the system. The message text will be deleted, but the reference information will be kept.

`KeepMsg On/Off`
: Toggles the KEEP flag on a message so that this message will not be deleted when packing the conference. If the KEEP flag is already set, this option will clear this flag.

`DeleteMsg On/Off`
: Toggles the DELETE flag on a message so that this message will be shown as deleted/not deleted. Once you leave the conference the message can no longer be undeleted. Use this function with caution!

`SuperUnread On/Off`
: Toggles the SUPERUNREAD flag used to avoid a message being marked as read. Neat for important messages.

`CatchUp`

`All`
: Mark all messages in current conference as read.

`Upto`
: Mark all messages upto the current messages as read.

`Pack/Purge Msgs`
: Opens the [Pack/Purge Window], page 8.

`Move Messages`
: Will let you move the current or all select messages in the message list window to another conference. If the conference do not exist, it can be created by typing in a new name in the string gadget.

`Show >>`
: Will let you select the read mode to read messages in. See [Show], page 12.

`Group by >>`
: Will let you select which method is to be used to group messages. See [Group by], page 12.

`Windows`

`Conflist`
: Opens the [Conference Window], page 14 or closes it if it is already open. Not available from the startup window.

`Msglist`
: Opens the [Messagelist Window], page 13 or closes it if it is already open. Not available from the startup window.

`Dock`
: Opens the [Function dock], page 10 or closes it if it is already open. Not available from the startup window.

`File Database`
: Opens the [File database], page 15 window or brings it to front if it is already open. Not available from the startup window.

`User Database`
: Opens the [User database], page 17 window or brings it to front if it is already open. Not available from the startup window.

`Kill Database`
: Opens the [Kill database], page 18 window or brings it to front if it is already open.

`Newfiles`
: Opens the [Newfiles Window], page 10 or brings it to front if it is already open. Not available from the startup window.

`Marking of msgs`
: Opens the [Mark Window], page 23.

`Events Editor`
: Opens the [Event Window], page 21 or brings it to front if it is already open.

`Event Commands`

`Script Command`
: Will let you enter a string that will be sent to the system by the script. This allows for custom commands to be sent whenever you want to automate an operation. This option is only available if the system type supports this event.

`Get Newfiles`
: Will request a list of new files from the system the next time you call. This option is only available if the system type supports this event.

`Autologoff`
: Lets you choose whether you want the script to log you off automatically from the system after the script session is ended. This option is only available if the system type supports this event.

`Retract File`
: Lets you retract a file on the remote system. This option is only available if the system type supports this event.

`Cancel Message`
: Lets you retract (cancel) a message/article on the remote system. This option is only available if the system type supports this event.

`Get Conflist`
: This option will request a list of all conferences from the remote system. It's typically used for getting a list of newsgroups. This option is only available if the system type supports this event.

`Download`
: Lets you enter a filename for a file to be obtained from the remote system. This option is only available if the system type supports this event.

`Upload`
: Will open a window with options that will let you upload a file to the remote system. See [Upload], page 11. This option is only available if the system type supports this event.

`Resign Conf`
: This option will let you select a conference to resign (drop) from the conferences you receive messages in. This option is only available if the system type supports this event.

`Join Conf`
: This option will let you select conferences to join so that you will start getting messages in these conferences. This option is only available if the system type supports this event.

`Send Userinfo`
: This option will let you enter some information about yourself that will be entered on the remote system. This option is only available if the system type supports this event.

`Configure System`
: This option will configure some parameters for you on the remote system. This option is only available if the system type supports this event.

`Forward Message`
: This is used to forward the current message to another user. This option is only available if the system type supports this event.

`ARexx Scripts`

`Add Script`
: This option will let you choose a ARexx script to put on a given function key. Several useful scripts are included in the THOR package for this purpose.

`Edit Definition`
: Will let you edit a macro previously assigned to a function key.

`Save Definitions`
: Will save your macro definitions to the file 'THOR.macros' as pure ascii text.

`Execute Script`
: Will let you execute an ARexx script which isn't assigned to a function key. A file requester will come up, and you can choose a script file from this requester.

`F1 - F20`
: This is where your macros will show up. To execute a macro, you can choose the macro from the menu with the mouse, or press the corresponding function key (F11 = SHIFT + F1.)

7 System types and formats supported

System types supported in Thor:

Please give us suggestions for other system types you would like to see implemented in THOR. We are very interested in any format specifications. If you send us this, we might add support for it.

The ABBS/MBBS system type

THOR has been extensively tested against these systems, and should work with a high degree of reliability.

Please note that you must set the character set in your NComm to ISO if you are using THOR against ABBS, and to IBN if you are logged on to MBBS. If you will be using the supplied NComm script, this is not needed.

You must also be sure that you have chosen English as language on MBBS if you intend to use THOR and the included script with MBBS.

The QWK system type

When registering a new system, you can set the Type to QWK. This is, however, a system type with no prepared scripts, so you either have to log on without a script, and get the QWK packet yourself or edit the script to adapt it to the system you are setting up. Please see the Chapter 8 [Scripts and communication], page 42 chapter for information on how the scripts interact with THOR.

Note that if your system has an option to turn *.NDX and *.PTR files off, please do so. THOR does not make any use of these.

The Hippo system type

When registering a new system, you may set the Type to Hippo. This is a packet format used on BBBS systems. If both QWK and Hippo is available, Hippo is preferable since it does not have the 25 character subject, to and from limits that QWK suffers from.

Please note that the Hippo format supported is version 2 of the Hippo standard, so make sure that you choose this format on the system.

BBBS is supported with a script for NComm.

The BlueWave system type

When registering a new system, you may set the Type to BlueWave. This is a very good and advanced format that is popular especially in the USA. If your system supports both QWK and BlueWave it is suggested that you use the BlueWave format. Some of the advantages when compared to QWK are:

-Automatic flags for the conferences. This means that THOR knows
about read-only conferences, which conferences you have subscribed etc.
-Longer subject, to and from-strings.
-Aliases.

The Omen system type

When registering a new system, you may set the Type to Omen. Omen is a fairly rare format used on BBSes in Finland mostly. This format has not been tested all that much, so any feedback is appreciated.

The SOUP system type

When registering a new system, you may set the Type to SOUP. SOUP is short for 'Simple Offline Usenet Packet'. This is a format that has become very popular lately, mainly because the newer versions of UQWK for UNIX systems supports it. If you have access to a dial-up UNIX account and would like to read News and E-Mail at home, check out uqwk.

The SOUP format has a lot of advantages over the QWK format, especially when reading News and E-mail. Subjects and adresses are not limited to 25 characters and in UQWK the threading actually works (unlike in QWK mode) The SOUP format is highly recommended over the QWK format if possible.

AmiTCP users may use the SOUP format by telnetting to the internet provider and run uqwk there, and then transfer the resulting packet via ftp. Or use the [TCP], page 35 module, which is slower, but a bit more comfortable.

THOR creates reply-packets with the name <mail-packet name>.out

Note that there is a special UQWK_SOUP type. This exists because uqwk does not quite follow the SOUP standard, so you have to use this type instead of plain SOUP if you want resigning and joining of newsgroups to work from the GUI.

Along with the THOR distribution is a script that is meant to be used on the remote UNIX system for grabbing news and mail using uqwk with SOUP. This script is located in the unix/ directory. Upload this script to the remote UNIX machine and type the following to make it executable:

chmod +x grab

If you want to use some other mail packet name than 'soup', change the script accordingly. A plain 'grab' suffices normally, but some options are available. An alias for the options you want is a good idea.

The options available in the script are:

-r : Do not use the "rz" program to transfer replies (<mail-packet name>.out) to the UNIX side.
-s) Do not use the "sz" program to transfer the grab from the UNIX side.
-n) Do not grab news. Nice if you just want to read mail and do not have time to grab news.
-m) Do not grab mail. If you for some reason do not wish to read new mail, use this. If you just want to enter replies, '-n -m' works fine.
-u) Do not limit the size of news grabbed to 400kB.
-x) Restore mail, used if you somehow lost mail the last time you grabbed.
-z) Restore news, used if you somehow lost news the last time you grabbed. '-x -z' works fine.

UQWK_SOUP is supported with a simple NComm script, that might or might not work on your Unix system (modifications might be needed to make it work correctly).

The FIDO system type

To use FIDO with THOR, set up a new system with system-type set to FIDO. It is suggested that you configure a separate upload directory for the FIDO-system, since FIDO tends to make several files there.

If you have used SPOT/Trapdoor before, you already have Inbound/Outbound directories set up, so just set the download/upload directories to these directories (In Files/Paths).

FIDO-packets are usually named <something>.PKT or <something>.<weekday><grabnr>, where <something> is a 8 digit hexadesimal number, <weekday> is one of MO, TU, WE, TH, FR, SA and SU and <grabnr> is a number between 0 and 9.

Therefore, this pattern for the Mail packet name will cover most, if not all, names a packet may have:

"#?.((MO|TU|WE|TH|FR|SA|SU)[0-9]|PKT)"

(Write this, without quotes, in the Mailpacket string gadget)

Next, set up a Email address in the User Options window (User Info). All references to Email address in THOR, means your *FIDO* address, so this is your net-address (For instance, my

address is 2:212/8.15) If you're not a registered user of THOR, or you use another name on the FIDO system than is in your keyfile, you must also type in your username.

After that, go to the Import/Export window. Here you kan set up the archiver to use when archiving events. Default is Zip, and you will probably never need to change it. You can also set up which charset to use when writing messages. Default here is ISO (LATIN-1). If your boss doesn't like that you write 8-bit messages, change to one of the 7-bit charsets (NO7, DK7, UK7 etc..). This charset is also used when importing messages, if the messages does not contain a CHSR/CHARSET kludge.

The other things in the Import/Export window are explained in the THOR documentation.

Type Options next. This is an ARexx script, which let you set some FIDO options.

Origin: You may customize the Origin line, which is added to all the messages you write (except netmail-messages). If you specify "My Point" here (and your fido-address is 1:2/3.4), the messages will get the line " * Origin: My Point (1:2/3.4)" added to them. If you don't specify an origin text, the name you configured for your FIDO system will be used instead.

Password: This password will be used in all your reply-packets.

Force INTL lines: If you experience problems with your netmail not coming to the receiver, try setting this to yes.

Pack NetMail: if you write many messages in NetMail.*, it could be an idea to pack these messages to improve transfer-time when uploading packets to your boss. If you write one or two messages each time, it probably won't help much.

UUCP address: This is the address of the gateway to send internet messages through (will be explained later).

UUCP name: This is the name of the UUCP gateway.

`Uploading you reply-packets`
: Since THOR only parses message packets, and makes reply-packets of your messages, you need an external program to download/upload messages from/to your boss. Trapdoor works ok.

`Multiple nets.`
: THOR supports multiple nets, that is, you can parse packets from different nets into one system. To accomplish this, THOR automatically configures an EMail address for each conference (AREA) when adding messages to this conference for the first time. After this, when writing messages in this conferece, the conference EMail address is used instead of the global one. Also, when receiving messages in these conferences, a Netmail-conference is automatically created for each of your FIDO-addresses.

`Writing UUCP-messages`
: By configuring UUCP address and name in Type Options, it is possible to automatically route UUCP addresses through an UUCP gateway. If the address of the gateway is 1:100/200.5, and the uucp-name is UUCP, a message to john.smith@someplace.somewhere will be sent to UUCP at 1:100/200.5, and the fist

line in the message will be "To: john.smith@someplace.somewhere". THOR will automatically recognice john.smith@someplace.somewhere as an EMail address, but the recognition routine is very simple, so all addresses not beginning with a number, will be recogniced as an EMail address. This also means that it is not possible to send messages to EMail addresses beginning with a number. This will be fixed in a later release.

`Downloading files: FREQ`
: To request files from your boss, use Download and type in the filename as you would have done with any file requester.

`FileAttachments`
: To attach a file to a message, use the Attach File in the Extras window of the enter/reply window. When attaching a file to a message, the subject you write is ignored.

`AreaFix`
: There is at this time no support for areafix, so joining and resigning AREAS must be done manually. Sorry.

`FIDO_EXPORT`
: If you don't want to have to delete the events after you have sent them, you can use this type instead of the normal FIDO type. When using this type, all events are exported only once; when you leave the fido-system. The drawback with using this type is that you won't be able to edit the events after you have exited the system.

  If you want to be able to edit your events, you could remove the eventpacker, and insert a line in your call-script (if you use one).

  Run the command run ‘getenv THOR/THORPATH‘cfgtype FIDO_EXPORT NEP to remove the eventpacker, and insert a line like

  <path to your thordir>bin/packfido <systemname> EXPORT PUBSCREEN ""

  in your call-script. This line exports all events.

The UUCP system type

When registering a new system, you may set the Type to UUCP. The grabname should be set to the name of your incoming mailbox.

If you want to use THOR as a newsreader as well as a mail reader, you will have to run the type config in the Systems configuration. Also remember to configure your email address in the Systems configuration.

This document will not explain how to install UUCP. This is a non- trivial task, and beyond the scope of this manual. We suggest that you read the manuals in the UUCP archive carefully.

You should always run the UUCP Type options before you start using a UUCP system in THOR. You will be queried for the following information:

`Headers`
: By selecting Headers you can edit a file that always will be inserted in the beginning of the mail/postings you write. This is done before the editor is started so you have the chance to edit the headers along with the rest of the mail.

This feature is nice if you always want to have some special information in the header of your mails, such as a Organization: header or a Reply-To: field. You may also write headers at the top of the mail you are composing even if you don't use a header file.

`Configuration`
By selecting Configuration you will be prompted for various questions that changes the behaviour or the UUCP module.

`Use THOR as a news and mail or mail only reader?`
Should be self-explainatory. If you choose mail only, your mailbox file will be processed by THOR, not the files in the news tree.

`Select news directory`
If you chose to use THOR as a news reader, you will be prompted for the root directory of your news tree, where your news is stored. This is normally UUNews: which is the default.

`Should THOR use a From: line in postings?`
Some UUCP implementations wants a From: line in the header of the file that postnews is delivered. Try this first, and if it doesn't work, run the configuration again and change it not to add a From: line. If you are not adding a fromline you will have to use arguments the sendnews command, see below.

`News posting command`
Normally, a command called postnews is used to post news articles. This is the default choice. If you have configured no From: header in the previous option, you will have to add the sender on the command line to the postnews command. An example of such a command would for me be

postnews -f kjelli -r "Kjell Irgens"

If this doesn't work, read the documentation of the postnews you use carefully.

`Should THOR use a From: line in mails?`
Some UUCP implementations wants a From: line in the header of the file that sendmail is delivered. Matt Dillons implementation does not want a From: line while most other do. If you are not adding a fromline you will have to use some arguments to the sendmail command, see below.

`Mail sending command`
Normally, a command called sendmail is used to send off or spool mail. This is the default choice. If you have configured no From: header in the previous option, you will have to add the sender on the command line to the postnews command. An example of such a command would for me be

sendmail -f kjelli -R "Kjell Irgens"

If this doesn't work, read the documentation of the sendmail you use carefully.

When you have finished the config, you are ready to use the supplied programs for news and mail handling, SendUUCP and ParseUUCP.

Use the SendUUCP command from the bin directory in the THOR drawer to send off your email and postings. This command calls sendmail and postnews on the events in THOR:

senduucp <mysystemname>

Run the ParseUUCP command from the bin directory in the THOR drawer to parse news and mail into THOR:

parseuucp <mysystemname> [mymailbox] DELETE.

This should parse the news tree and your new mail into Thor's database, and delete the UUCP format news and mail from your harddisk.

The TCP system type

These software tools provide an easy way receive and send news/mail using a TCP/IP connection (like AmiTCP, AS225, MLink, etc). Tools are provided that should satisfy both shell and GUI users.

All the TCP programs will work both under AmiTCP and AS225r2 (and compatible TCP/IP stacks). The current TCP/IP stack in use will be automatically detected and used.

GetTCP is used for getting mail and news, while SendTCP is used for sending mail and news that you have written, joining and resigning newsgroups, etc.

ConnectTHOR is used for integrating these programs into a single program. It also contain a nice news browser what will let you select a newsgroup and download only the articles you want be selecting them from a list with dragselection.

GetTCP and SendTCP can both be started from shell with arguments that will override the configuration settings. If you want, you can run these automatically from eg. a startnet script.

Starting with THOR 2.3, ConnectTHOR and CfgTCP now use the ClassAct BOOPSI layout system.

Getting started with the TCP Internet modules

- Enter Systems Config in THOR and create (or change an already existing system) a system with Type set to TCP. Configure you email address in the "Userinfo" window. As "Mail packet", you can enter your username, and if you have mailpackets in your download dir, they will be parsed. Otherwise, you do not have to enter anything here.

- Click on the "Type Options" gadget, a window containing various options will come up. See [CfgTCP], page 38.

- When you have configured the various options described above, you must use the "join" option from the events menu or the dock window in THOR to select which conferences to collect news from. This is required even if you received several groups previously with UUCP or SOUP. To get a complete list of newsgroups from your newsserver, select Get Conf.List from the Dock window or the Events menu, then use "Send Events". "Send Events" is also required when you have joined newsgroups.

The ConnectTHOR Internet module

ConnectTHOR is used for integrating CfgTCP, SendTCP and GetTCP into a single program. It also contains a nice news browser what will let you select a newsgroup and download only the articles you want by selecting them from a list with multi-selection. The options available are:

`Do All`
: This option will send pending events you have created, then collect new mail and news, all in a single operation.

`Get Mail`
: This option will collect mail from your configured POP3 mailserver.

`Get News`
: This option will collect news from your configured NNTP newsserver.

`Send Events`
: This option will send pending events (eg. news and mail postings you have written).

`Config TCP`
: This option will let you configure TCP options using the [CfgTCP], page 38 program.

`Browse News`
: Selecting this option will make ConnectTHOR connect to your configured NNTP newsserver, and then open a window that will let you interactively browse newsgroups. The following options are available in this window:

  `Newsgroup`
  : This string gadget will let you enter manually a newsgroup to connect to. When you press return after having entered the name, ConnectTHOR will connect to this group on the newsserver and update the slider gadgets accordingly. If you enter a newsgroup not present on the newsserver, ConnectTHOR will warn you about this.

  `Group List`
  : If you select this option, a window containing all newsgroups on the newsserver will come up. (NOTE: If you haven't collected a list of groups using GetConfList in THOR and then Send Events, only newsgroups you have previously received messages in will come up here). You can then select a newsgroup to browse.

  `Group Hotlist`
  : ConnectTHOR's newsbrowser maintains a list of newsgroups you have previously browsed in an internal list, that will be saved to disk when you exit the browser. The message pointer will also be saved. By selecting this option. you will be able to select a newsgroup from this "hotlist" of newsgroups. NOTE: This hotlist is NOT connected to the subscribed newsgroups in THOR in any way.

  `From`
  : This is from-message number that will be used when collecting the headers.

  `To`
  : This is to-message number that will be used when collecting the headers.

`Get Headers`
: This option will collect some information from the message headers in the current newsgroup and shown them in the listview in the browse window. What information you can collect depends on the newsserver. On most newsservers, you will atleast get messages numbers and subjects. However, INN and compatible newsservers supports sending more information. On these type of servers, you will also get the messages sizes and who wrote the message.

`Get Articles`
: When you have selected the articles you want, you can use this option to collect these articles from the newsserver.

`Clear/Mark`
: With clear or select all entries in the listview, depending on if there are selected entries or not.

`Group by`
: Will let you group the collected headers by MsgNr (default), Subject or Author.

`Edit Hotlist`
: Will let you remove entries from the hotlist maintained by ConnectTHOR.

`Filter Crossposting`s
: This option, when enabled, will make sure you only see crossposted articles once in the current browse section.

ConnectTHOR is also a commodity and supports the following tooltypes and command line arguments:

`SYSTEM`
: This is the configured system in THOR that will be used to collect news and mail into, and send mail and news from.

`PUBSCREEN`
: This is the public screen that ConnectTHOR will use when first started.

`CHECKDELAY (PERIODIC)`
: This is the number of minutes between each time you want ConnectTHOR to collect mail for you.

`IMMEDIATE`
: Will make ConnectTHOR collect new mail on startup automatically.

`BROWSENEWS`
: Using this argument will make ConnectTHOR open the news browser directly on startup.

`NEWSGROUP`
: This argument is used in cooperation with BROWSENEWS, and contains the name of the newsgroup to get the headers for.

`CX_PRIORITY`
: This is the priority ConnectTHOR is to have in the commodities input events chain.

`CX_POPKEY`
: This is the shortcut that must be used to pop ConnectTHOR to front if it's hidden.

`CX_POPUP`
: Set this to YES if you want ConnectTHOR to open its window each time you start it.

The Config TCP Internet configuration program

`Misc`

`Logfile`
: Full path and name of optional logfile to use for logging sending and receiving of mail and news.

`No logging`
: Logging will not be done when this gadget is checked.

`EMail Address`
: This is your email address.

`Real Name`
: This is you full real name.

`Edit Headers`
: This option will let you edit custom header fields to be used when posting news or mail from THOR. Fields can include "Reply-To: <address>", "Organization: Not organized", etc. If THOR is running when you select this option, the editor configured in THOR will be used. Otherwise, a name of an editor to use must be entered in the requester that comes up.

`Delete Header`
: Will delete the custom header file for this system. Use this with caution since there is no way to get it back once it's deleted (except for using DiskSalv or somesuch :)

`Mail`

`Mail Server`
: This is the server that will be used when receiving mail (and sending mail, unless you have configured another server as SMTP Server for sending mail). Typing in the name or IP address of your pophost/mailserver will do.

`Delete mail from POP server`
: Checking this item will make ConnectTHOR (GetTCP) delete the mail from the remote POP server after the mail has been successfully collected. You might choose to let the mail remain at the remote POP server, in which case only the mail that is new from the last time will be collected.

`SMTP Server`
: This is the optional SMTP server used when sending mail. Normally, you do not need to configure this and can use the Mail server above for both collecting and sending mail.

`Loginname`
: This is the name you log on to your account with.

`Get from EMail address`
: By setting this checkbox, the login name will be resolved from your configured email address.

`Password`
: This is an optional password. By setting the "Ask every time" option, you will be prompted for the password before collecting mail. This is the best solution for security reasons.

`POP Port`

If your pop server is on some other port than the default 110, set this to the correct port.

`News`

`Newsserver`

Enter the name or IP address of the newsserver (NNTP server) you will use to get news from.

`NNTP Port`

If the newsserver you are using are running on some other port than the default 119, enter the port here. This should normally be kept at the default 119.

`Sockets`

This is the number of sockets (connections) to use when collecting news. By setting this to 2 or higher, you might get a 30-80% speedup when collecting news, depending on your connection, your diskspeed, your THOR setup and your newsserver. 3 or 4 sockets are generally good numbers to use. By using more than one socket, more of the available bandwidth will be used.

`Max Msgs`

This is the maximum number of articles that is to be collected from each newsgroup you are subscribing to. If you choose to use this option, you might miss some articles, but using it can be usefull when you have been away or somesuch, and do not want to collect all articles since the last time.

`NNTP Access Authentication`

`No NNTP Access Authentication (normal setting)`

If your newsserver do not require username and password, check this option. Normally, authentication is not needed.

`NNTP UserID`

This is the user id you have been given to have access to the newsserver, if it requires authentication.

`NNTP Password`

This is the password you have been given to have access to the newsserver, if it requires authentication.

`Aminet`

This window is used for configuring the ftp/Aminet support added in THOR 2.4.

`Prefered Aminet site`

This is the Aminet site you would prefer to use for uploading or downloading files. Note that the base directory can be changed, but normally it should be left as it is.

`Download dir`

This directory will be used for storing files collected.

`Get .readme files`

Selecting this option will cause THOR to get the corrensponding .readme file for each file.

`Save`

Save the configuration. If you start this program from ConnectTHOR, ConnectTHOR will update its internal variables to reflect on the configuration saved.

`Cancel`

Cancel the changes and quit the CfgTCP program.

The SendTCP Internet module

SendTCP is the shell program spawned by ConnectTHOR when sending mail, news and other events. FTP is also handled by SendTCP if you have pending upload or download events. Usually, this is the method prefered for doing this, but SendTCP can also be started from shell. The following arguments is understood and will override any setting configured in CfgTCP:

`BBSNAME`

Name of the system configured in THOR to send events from.

`MAILSERVER`

This is the name or IP address of the SMTP server to use. Usually, this is also the name of the POP3 server, but any known SMTP server can be used here.

`NEWSSERVER`

This is the name or IP address of the newsserver (NNTP server) to use when posting news replies.

`NNTPPORT`

If the newsserver you are using are running on some other port than the default 119, enter the port here. This should normally be kept at the default 119.

`PUBSCREEN`

If this argument is provided with the name of a valid public screen, SendTCP will open a progressbar on that screen.

`NNTPUSERID`

This is the user id you have been given to have access to the newsserver, if it requires authentication.

`NNTPPASSWORD`

This is the password you have been given to have access to the newsserver, if it requires authentication.

`LOGFILE`

Full path and name of optional logfile to use for logging sending of mail and news.

The GetTCP Internet module

GetTCP is the shell program spawned by ConnectTHOR when collecting mail and news. Usually, this is the method prefered for doing this, but GetTCP can also be started from shell. The following arguments is understood and will override any setting configured in CfgTCP:

`BBSNAME`
: Name of the system configured in THOR to collect news and mail into.

`PUBSCREEN`
: If this argument is provided with the name of a valid public screen, GetTCP will open a progressbar on that screen.

`PASSWORD`
: This is the password to use then collecting mail from the POP3 server. Usually, this is the same password you use when you login to your account. If no password is provided here, GetTCP will ask for a password before collecting mail.

`MAXMSGS`
: This is the maximum number of articles that you want to receive in each newsgroup.

`MAXDAYS`
: Not used anymore, only kept for backwards compatibility.

`NOMAIL`
: Do not collect mail.

`NONEWS`
: Do not collect news.

`MAILSERVER`
: This is the name or IP address of the POP3 mail server to use when collecting mail.

`NEWSSERVER`
: This is the name or IP address of the newsserver (NNTP server) to use when collecting news.

`PERIODIC`
: This is the number of minutes between each time you want GetTCP to collect mail for you.

`SOCKETS`
: This is the number of sockets (connections) to use when collecting news.

`USERNAME`
: This is the login username to use when collecting mail.

`POPPORT`
: If your pop server is on some other port than the default 110, set this to the correct port.

`NNTPPORT`
: If the newsserver you are using are running on some other port than the default 119, enter the port here. This should normally be kept at the default 119.

`NNTPUSERID`
: This is the user id you have been given to have access to the newsserver, if it requires authentication.

`NNTPPASSWORD`
: This is the password you have been given to have access to the newsserver, if it requires authentication.

`LOGFILE`
: Full path and name of optional logfile to use for logging sending of mail and news.

`GETONLY`
: Only collect news and/or mail, do not add it to the database.

`PARSEONLY`
: Only parse already collected news and/or mail into the database, do not get any mail and news.

`DELETE`
: By setting this option, the collected mail will be deleted from your mail (pop) host after it's been successfully received by your Amiga.

The TCP_ONLINE Internet system type

The TCP_ONLINE type may be used if your Amiga is connected to the net at all times. The difference from TCP is that messages you write are automatically sent when you leave the system. Used along with the PERIODIC argument to ConnectThor or gettcp, sending and recieving of mail and news does not require any interaction with ConnectThor or sendtcp/gettcp.

8 Scripts and communication

Like other offline readers, THOR is made to save online time. Obviously, you save most time when your logons are automated with scripts.

When you create a system, a little script in the NComm script language is created in the Scripts/ directory. This script will look something like this:

```
SET $BBSName = "Ultima Thule"
varfile "ENV:THOR/THORPath"
readvar $THORPath
varfile close

CHAIN $THORPath"scripts/<TYPE>.NComm.script"
```

TYPE in the above script is the system type to defined for the system in the Systems Configuration in the ConfigTHOR program.

Using NComm with the BBS modules

The scripts supplied with THOR are made to work under NComm 2.0 or higher. These scripts are only for ABBS/MBBS, BBBS and UQWK_SOUP, and will not work as they are on other systems. However, if you want to write your own script for another communication program, we will be pleased to help you as far as we are able to. Scripts for other system types are also welcome.

Using other communication programs with the BBS modules

NComm 2.0 or higher is the only communication program supported with a script from us at this time. It should be possible to write scripts for other programs, but such scripts generally tends to be fairly slow and unusable on comm-programs with only ARexx as script language.

Please feel free to contact us for help on this if you want to write scripts for your favorite program.

9 THOR Full Screen Editor

The FSE in THOR is based on Pascal source from Mike 'MBBS' Robertson and converted to Amiga assembler by Marius Mortensen. It has also been improved in several aspects from the original editor in MBBS and the FSE in Offline. The editor has been maintained and upgraded for THOR 2.1-> by Petter Nilsen, due to Marius' workload at the place he works (Scala drives their programmers hard :)

The FSE is pretty self-explanatory (especially if you're used to using A/B/MBBS or the good ol' WordStar (ancient tool for writing garble)).

The FSE has extensive support for shortcuts See [FSE Shortcuts], page 44., a full menu with various options See Chapter 10 [FSE Menus], page 44.,

The FSE also has support for simple dos-scripts See [FSE DosScripts], page 47., aswell as full-fledged ARexx support. See [FSE ARexx], page 47.

The Full Screen Editor will multitask internally in THOR, since the editor is started as a separate cli process. THOR will communicate with the editor(s), so that Iconify and configuration related functions will have effect on the editor(s) too. An unlimited number of editors can be open at the same time (only limited by your free memory), and you can edit a message on one system while reading messages on another.

Each time you exit the FSE, THOR will save the position, size, insert/indent status and the XY display status.

Shortcuts in the FSE

```
 Enter  = new line                    Backspace  = delete prev char
 Home   = start line, page, message   Del        = delete current char
 End    = end line, page, message     TAB        = skip <n> columns

Ctrl/A back a word       Ctrl/B ins.deleted lines Ctrl/C forward a page
Ctrl/D forward one char  Ctrl/E up one character  Ctrl/F forward a word
Ctrl/G delete character  Ctrl/L toggle XY display Ctrl/N rewrite screen
Ctrl/O open new line     Ctrl/Q exit DON'T SAVE   Ctrl/R back a page
Ctrl/S back a character  Ctrl/T delete word       Ctrl/V insert/indent
Ctrl/W ins.last del line Ctrl/X down a character  Ctrl/Y delete line
Ctrl/Z SAVE and EXIT     Ctrl/K,R or W read/write file

Indent-status, insert-overwrite and whether you want coordinates shown are
remembered between sessions.

Up  to  30  lines  are remembered by Ctrl/Y and can be reinserted with
Ctrl/B.

Amiga-T - Inserts a random-tag from the Tag file defined in the
          ConfigTHOR program.
Amiga-V - Paste Clipboard contents.
Amiga-R - Do a ROT-13 on the section under the cursor in the text.
Amiga-M - Start/Stop learning macro.
F1-F20  - Execute macro.  (Note that macros do not nest).
```

Chapter 9 [THOR Full Screen Editor], page 43 - About the editor Chapter 10 [FSE Menus], page 44 - About the menus

10 FSE Menus

The FSE Project menu

```
About
-----
  Shows some very interesting info.

Include File... - Shortcut RAmiga-O or Ctrl-K/R
---------------
  Load a textfile or message and inserts it at the current
  cursor position.
```

```
Save As...      - Shortcut RAmiga-A or Ctrl-K/W
----------
  Save the message to a specified file.

Save/Quit       - Shortcut RAmiga-Z or Ctrl-Z
---------
  Save the message for upload to the system.  Note that the message-file
  itself may be removed after the upload.
  Then quit the editor.

Quit            - Shortcut RAmiga-Q or Ctrl-Q
----
  Abort without saving the message - a requester will ask for
  confirmation
```

The FSE Macro menu

```
Learn Macro     - Shortcut RAmiga-M
-----------
  Start/stop the learning of a macro.  Note that a macro cannot
  include another or itself.   A macro consists of keypresses.

Load Macros
-----------
  Load a file consisting of macro descriptions for the 20 f-keys
  The Editor attempts to load the file <YourThorDataDir>FSE.macros on
  startup.

Save Macros
-----------
  Save your macro-definitions.

Choose Startup-Macro
--------------------
  Lets you decide which macro to run at startup.
```

The FSE Miscellaneous menu

```
XY-Display      - Shortcut Ctrl-L
----------
  Toggle the coordinate-display in the upper right corner of
  the window.
```

```
  Status remembered by THOR.

2.0 Snapping
------------
  Toggle between cursor positioning by mouse-click, or standard
  2.0 console snapping.

  This setting is remembered by THOR.

Toggle Mode     - Shortcut Ctrl/V
-----------
  Change the Autoindent/indent and Insert/overwrite status.

  Remembered by THOR between sessions.

Change Subject  - Shortcut Ctrl-K/S
--------------
  Change the subject of the message.

  May not be available.

Toggle Private  - Shortcut Ctrl-K/P
--------------
  Change the private-status of the message.

  May not be available.

Rot13 Section   - Shortcut RAmiga-R
-------------
  Perform a ROT-13 on the section under the cursor.

Insert Tag      - Shortcut RAmiga-T
----------
  Ask THOR for a tag from the tagfile and insert it at the current
  position in the text.

Paste Clipboard - Shortcut RAmiga-V
---------------
  Will paste the contents of the clipboard.

Help            - HELP
----
  Shows general help for the editor.
```

The FSE ARexx menu

```
Execute Script...
```

```
-----------------
  Pops up a requester and lets you execute an ARexx script.

Edit Script Definition
----------------------
  Lets you change one of the ten pre-defined ARexx-scripts.
  Select this item, followed by the script-slot you want to change.

Save ARexx Defines
------------------
  Saves the ten script-slots to the file "FSE.Arexx".

"The ten slots" - Shortcut RAmiga-1 -> RAmiga-0
---------------
  Select one of these to execute one of the scripts.
  Note that if macro-learning is on, the script won't be run until
  the macro is actually used.  (This is done to prevent double
  output in the macro.)
```

FSE DOS scripts

Note that the editor has the possibility of letting dos-scripts process a message before and after the FSE gets its clammy filehandles at the text. The scripts are called FSEScriptIn and FSEScriptOut and are searched for in the directory you have chosen to contain your message database in.

The scripts get the parameters filename and width supplied as command arguments. This should be sufficient for quoting/adding of custom tags and so on.

NB:

Due to a bug in Toolmanager's WB-startup emulation code, the full dos- path will not be inherited by the scripts if you start THOR from TM in WB-mode. (The same goes for archivers, external editor and all other external programs started from THOR, of course.)

FSE ARexx

The editor supports these commands from ARexx:

```
XPOS           - returns cursor X-value.
YPOS           - returns cursor Y-value.
MAXMSGLENGTH   - returns maximum number of lines.
MSGLENGTH      - returns number of lines in this message.
LINELENGTH     - returns number of max-coloumns.
MSGFILENAME    - returns the full filename of the file in the FSE.
```

```
GETLINE [n]    - returns the contents of line number <n>, or the current
                 one if none specified..
                 returns RC=WARN if there are no lines in the message, or
                 the line requested is out of range.
QUIT           - Quit the editor without any requester.
REFRESH        - Make sure the window is redrawn.
SETPOS [X] [Y] - Move cursor to a specific position.
                 Note that 0 is a valid argument that does not affect the
                 cursor.   You may also leave out any arguments.

                  Examples:
                     SETPOS 10,10 - moves cursor to [10,10]
                     SETPOS 10    - moves cursor to position 10 on this
                                    line
                     SETPOS 0 10  - moves cursor to line 10 with unchanged
                                    X-value.
                     SETPOS Y 10  - Moves cursor to line 10 with unchanged
                                    X-value.

                 A WARN is returned if any of the coordinates are out of
                 range.
INSERTSTRING STRING/F
               - Insert the string supplied synchronously.  Will not be
                 noticed by i.e. macro-learning.
INSERTINPUT STRING/F
               - Asynchronous input-routine.  The string will be inserted
                 as if it were typed at the keyboard and may include
                 control-codes like CTRL-B, CTRL-QY etc.
PASTECLIP      - Paste clipboard contents at current cursor position.
INSERTTAG      - Ask THOR for a tag and include it at current cursor-pos.
SAVEFILE [NAME]
               - Save file.  If no name specified, requests one from user.
                 RC=WARN for error.
INCLUDEFILE [NAME]
               - Include file.  If no name specified, requests one from
                 user.
OVERWRITE ON/OFF
               - Changes the Overwrite/Insert status and returns current
                 status  Valid specifiers are ON=Y=YES and OFF=N=NO.
                 Returns ON or OFF depending on the current status.
AUTOINDENT ON/OFF
               - Changes the AutoIndent status and returns current
                 status  Valid specifiers are ON=Y=YES and OFF=N=NO.
                 Returns ON or OFF depending on the current status.
PLAYMACRO NUMBER
               - Play macro number 1-20.  Asynchronous.
DELETELINES [n]
               - Basically a CTRL-Y in a loop.
INSERTLINES [n]
               - Insert one or more empty lines at the current cursor
                 position (same as CTRL-O).
```

```
INSERTBUFFERLINES [n]
               - Insert lines from the buffer.  A CTRL-B in a loop.
SENDER         - Returns the sender of the message in the FSE.
                 (not yourself, but the one that wrote the message you're
                  replying to - if any.)
RECIPIENT      - Returns the receiver of the message you're writing.
                 (In a normal reply-situation, SENDER=RECIPIENT if it's
                  not forwarded.)
SENDERADDR     - Returns the senders address of the message in the FSE.
                 (not yourself, but the one that wrote the message you're
                  replying to - if any.)
RECIPIENTADDR  - Returns the receivers address of the message you're writing.
                 (In a normal reply-situation, SENDERADDR=RECIPIENTADDR if it's
                  not forwarded.)
CONFERENCE     - On replies, returns the original conference of the message
                 in the FSE.
SENDERDATE     - The date and/or time at which the message was written.
ISFIRSTCALL    - Returns:
                 "ON"  : This is the first time the editor is called on
                         this message (i.e. you may want to quote it.)
                 "OFF" : Edit message etc.
NEWLINE        - The same as a <return> press.
BACKSPACE [n]  - The same as a backspace-press.
BEGOFFILE      - Moves cursor to beginning of file (1,1)
ENDOFFILE      - Moves cursor to the position behind the last char in
                 the text.
BEGOFLINE      - Moves cursor to the first char on the current line.
                 (Note that this does not include whitespace).
ENDOFLINE      - Moves cursor to the position behind the last char on the
                 current line.
CURRENTCHAR    - Returns the character under the cursor.  Remember to
                 check for "".
HIDE           - Hides the editor.  The next ARexx-command sent will wake
                 it up again.
REVEAL         - A simple nop-command to wake the editor from hybernation.
NOP            - Do nothing.
REQUESTNOTIFY "TEXT/A,BUTTONTEXT=BT/A"
               - Asks THOR to show a requester,  See the ARexx-docs for THOR
                 for further details.
REQUESTSTRING "TITLETEXT=TITLE/A,BUTTONTEXT=BT/A,INITIALSTRING=ID,MAXCHARS/N/A"
               - Asks THOR to show a string-requester,  See the ARexx-docs for
                 THOR for further details.
GETFSEVERSION  - Returns the version of the editor.
GETFSEREVISION - Returns the revision of the editor.
FSESCREENNAME  - Returns the name of the screen this FSE resides on.
```

11 Message reading and writing

Several gadgets can be used for navigation through the database in THOR. These gadgets are present at the bottom of the main reading window (unless you have chosen to hide them by selecting "Hide main window button panel" under Screen Options in the global configuration part of ConfigTHOR):

`Reply`
: This will let you write an private answer to the current shown message (if any). See [Reply], page 54 for a more complete description of this function.

`Followup`
: This will let you write an answer to the current shown message (if any). See [Reply], page 54 for a more complete description of this function.

`Write`
: Does not differ from *Reply* except no reference text will be imported into the editor.

`Next`
: This function will go to the next message if there are any in this conference. If not, the next conference is entered.

`Prev(ious)`
: Works like the *Next* function except that it will go to the previous message.

`Next conf(erence)`
: If you are in All-mode, this will take you to the next conference. If you are in Unread-mode, which is the usual read mode, this will take you to the next conference with unread messages. In Search-mode, this will take you to the next conference with messages found in the message search.

`Prev(ious) conf(erence)`
: Works like the *Next conf(erence)* except that it goes backwards.

`Mark`
: Opens the [Mark Window], page 23 the allows you to mark messages in several different ways.

Reading of messages can be done by using the buttons at the bottom of the Main Window, but most often it's more convenient to use shortcut keys. See [Shortcuts], page 50.

Shortcuts when reading messages

The following shortcuts are available when reading messages (each shortcut is enclosed in double quotes):

```
"n" "+" "."        - Go  to  next message. If there are no more
"cursor right"       messages  in  the  current  conference,   a
                     window will pop up requesting which conference to
                     read next.

"b" "p" "-" ","    - Go to the previous  message.   If there are   no
"cursor left"        previous   messages   in  the  current conference,
```

a requester come up asking if you want to go to the previous conference.

"N" - Go to the next conference.

"P" "B" - Go to the previous conference.

">" "Shift Cur. Right" - Read the first answer to the current message. (Thread forward.)

"<" "Shift Cur. Left" - Read the message which the current message was the answer to. (Thread backwards.)

")" "=" "Alt Cursor Right" - Read the different answers to the message the current message is an answer to.

"(" "Alt Cursor Left" - Read related messages backwards from the current message.

"r" - [Reply], page 54 to the current message, make the reply default to the mail conference.

"f" - [Reply], page 54 to the current message.

"e" "E" - Enter a new message.

"g" - Resign (unsubscribe) from a conference. The current will come up as default in the requester.

"j" "J" - Join (subscribe) a conference.

"i" "I" - Send userinfo about yourself to the remote system.

"v" - Open the Event Window where you can do various changes to existing events on the current system.

"w" "W" - Change between viewing ALL, Unread or messages found in a search.

"o" "O" - Change between the various methods of sorting messages.

"G" - Order a full conference list. Will be done the next time you call this system.

"d" "D" - Download a file from this system. The file will be downloaded the next time you call.

"M" - Mark the current message as unread.

"m" - Open the Mark Window which will allow

```
                    you to mark messages in different ways.

"u" "U"           - Open the Upload Window where you enter
                    information about files to upload to this system.

"C"               - Catchup  will  mark  all  messages in the current
                    conference as read.

"a" "A"           - Will let you pack and purge all or the current
                    conference.

"ENTER"  "RETURN" - Scrolls the current message one line towards
                    the end.

"SPACE"           - Advances the text in the current  message one
                    page forward.   If the last line in  the  message
                    text is shown, this will go to the next message.

"ESC"             - If the Main Window is active, this will close the
                    system and return you to the Startup Window.
                    If the SHIFT key is also pressed, no outgoing
                    event packages will be created.
                    If any other window is selected, that window will
                    be closed.

"TAB"             - Cycles between and, if shift-backspace is used, brings
"BACKSPACE"         to front the different THOR windows.

"CTRL Cursor Up"  - Will bring you to the top of the  message text.

"Ctrl Cursor Down" - Will bring you to the bottom of the message text.

"HELP"            - Will give you help on the window under the cursor.

"`"               - Will turn on/off viewing of the extra header fields
                    present on news/mail and Fido systems.

5 (num keypad)    - Mark messages with the current subject as read.

Numerical keypad  - Will work as described on the keys (eg. Home etc)
```

See Chapter 11 [Message reading], page 49, and Chapter 4 [Windows in Thor], page 6 for more info.

All thread functions requires that the system type supports reference information.

Please note that the availability of the different functions described here will differ from system to system. In that case, selecting an unsupported function will not have any effect and the appropriate gadget will be ghosted.

The message header

The message header is used to display important information about the current message. The header is located at the top of the Main Window and will show the following information when reading messages:

`From:`

The sender of the message. If any address is available, it will be enclosed in <> brackets after the name.

`To:`

The receiver of the message. If any address is available, it will be enclosed in <> brackets after the name.

`Date:`

Date and time when the message was written.

`Subject:`

The subject of the message.

`ReplyTo:`

The message number that the current message was an answer to, or "NONE" if the database contains no reference to a previous message in the thread.

`OrgNr:`

Original message number from the remote system, if any.

`Replies:`

The number of replies this message has.

`Flags:`

The following message flags can be present here:

P - The message is private.
R - You have replied to this message.
K - The message is marked with the KEEP flag.
D - The message is deleted from the message database.
U - The message is unread.
I - The message is either marked as Important or as Urgent
S - The message is marked as SuperUnread.

In the top right corner, to the right of the message header, is two small buttons.

The first one at the top is for turning on/off the viewing of the extra header fields that might be present on some system types. Currently, only Fido and news/mail systems supports this.

The second one is for turning on/off style tags support in THOR. Please see [Visual Options Window], page 6 for a description on what style tags are.

Reply to a message

This is used when you reply to the message shown in Main Window, Forward or when you enter a new message. The following options are available:

`To:`
: Here you can enter or change the receiver of the message. If you enter a name that is not in the user-database, a window will pop up with suggestions of names based on what you entered. Click on the name you want or use the up and down arrows to selected a name and press enter to have it inserted into the To: field. ESC will leave the name as it is.

  This field may not be available, depending on the system type.

`To-Address:`
: If the system type uses addresses instead of or in addition to names, this field is available.

`Subject:`
: Here you can enter or change the subject of the message.

`Conference:`
: Here you can change the conference the reply should be entered into. THOR will try to fill out the complete conference name if you do not enter a complete name of the conference.

  The conference can also be selected from the list of conferences that pops up if you click on the button to the right of the field.

`Private:`
: With this you can change if the message is to be Private or readable by all. Not all systems and conferences allows you to change this status.

  This field may not be available, depending on which system type the current system uses.

`Reflow:`
: By deselecting this tag, the original message is not re-formatted in the editor. This is useful on some special messages where the reflow system gives an unwanted result.

`Include Text:`
: Option for letting you include the text of the message you write an answer to or not.

`File Attachment`
: If the system type supports it, you can select a files to include with the message. On Internet news/mail type of systems, this files will be coded using the base64 method and sent in a multipart MIME message. Other system types might use other methods for sending this file to the receiver.

`Extras...`
: This window contains other options available when writing or replying to a message.

  `PGP Encrypt`
  : Selecting this option will let you encrypt the message with the given receivers PGP public key.

  `PGP User ID`
  : This is the user id from the keyring to use when encrypting a message using PGP. This is NOT the key id, which is something else!

`PGP Sign`
: Selecting this option will sign the message with your secret PGP key.

`Urgent`
: If the system type supports it, you can set this flag to signify that this message is important and should be given high priority. On Fido systems, this is the same as "Crash mail".

`Receipt`
: If the system type supports it, you can set this flag to signify that you want a receipt message from the receiver when he receives this message.

`Encode as 7-bit`
: If the system type supports it, you can set this flag to signify that you want this message to be coded for transmition over a 7-bit (that strips national characters etc.) On Internet news/mail system types, this means that the message will be encoded with a MIME method called "quoted-printable" (or quoted-unreadable as some like to call it).

When you are satisfied with this you press return (with no active gadgets) or click on "Ok".

12 MIME support in THOR

What is MIME? (from RFC1341)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

MIME stands for "Multipurpose Internet Mail Extensions". It is the standard for how to send multipart, multimedia, and binary data using the world-wide Internet email system. Typical uses of MIME include sending images, audio, wordprocessing documents, programs, or even plain text files when it is important that the mail system does not modify any part of the file. MIME also allows for labelling message parts so that a recipient (or mail program) may determine what to do with them.

THOR has extensive MIME support, and will automatically handle MIME on incoming messages, with internal support for:

text/plain
text/richtext
text/enriched
message/rfc822
message/partial
multipart/mixed
multipart/alternative
multipart/digest

It will also automatically decode the two major encoding schemes:

quoted-printable
base64
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

"Multi-media" mails with binary attachments will get an icon in the main window listview. This icon can be clicked on to perform certain operations on the attachment, including saving. When you have the appropriate datatypes installed, you can view pictures within the text and playback sound. The icons that is shown in the listview for each attachment are loaded from the filetypes/ directory in the main THOR dir. You can replace them with your own icons if you want, or rename one of the other filetypes#? directories to get different icon set. The icons should be project icons, and the default tool of the icons will tell THOR what external viewer to use, if the user chooses to use an external viewer.

You can also attach files to outgoing messages in Thor. The attached files are encoded using base64. The message text can be encoded with Quoted-printable.

Attached files received by you will get an "icon" in the main message area listview. Clicking on this icon will allow you to perform various operations on the attached file, depending on what type it is.

13 Features for the Power User

This section is an overview of features that are hidden from the user most of the time, but still offer a lot to the way THOR functions. Most users probably won't need any of these features mentioned here, but experienced users might find them useful.

`Selecting conferences from a conference string gadget`

Several methods can be used to select one or several conferences to post a message to. In this example, let's say we have 3 conferences:

```
comp.sys.amiga.datacomm
comp.sys.amiga.applications
comp.sys.amiga.announce
```

Now, by entering just "c" and then pressing the Return key, all conferences starting with "c" in their name will come up in a list where one or several can be selected. In this case, all conferences will match.

If you enter "c.s.a.d", only one conference matches the pattern, namely "comp.sys.amiga.datacomm", so this conference will replace the pattern entered into the string gadget.

What THOR does when entering "c.s.a.a" is simply to expand this string to "c*.s*.a*.a*" and then perform a pattern match on all conferences in the conference list on the current system.

However, if we have conferences separated with a slash, like "THOR/Support", the short way to enter this conference' name would be "t s".

Using the cursor up/down keys while a conference string gadget is active will cycle through the available conferences, starting with the conference matching the pattern in the gadget.

Entering "*" will open a window with a list of all available conferences on the current system.

`Crossposting to several conferences`
: By using the methods outlined above, a window with a list of conferences can open and let you select several conferences to post a message to. Selecting several conferences in this window can be done by using the cursor up/down keys and pressing space for each conference, or by using multi selection with the mouse.

  The selected conferences will appear in the conference string gadget and the message will be posted to all these conferences. The method used for crossposting can differ from system type to system type. e.g. on QWK systems, THOR will create one event for each conference, while on SOUP and UUCP systems, one event will contain all conferences.

`Crossposting to several users`
: By using close to the same method outlined above, it is likewise possible to send a message to several users. The users can either be entered manually separated with a ',' (comma) character, or by using the popup gadget and multi selecting the users from the list in the window that pops up.

14 External utilities

Several utilities are included with THOR, including:

ThorStat

```
950104
                              ThorStat V3.x

      Requires an Amiga with OS compatible with Workbench 2.04.
            Requires also Magic User Interface installed.

  Copyright  1993-1995 Ultima Thule Software, All Rights Reserved.
               Author: Eivind Nordseth & Petter Nilsen

ThorStat
~~~~~~~~
Welcome  to  the first ever utility for the Thor system.  This utility
counts  messages in your database, it also counts messages to and from
you.  Which  systems  and  conferences  to  count  can  be selected in
listviews.  It is quite simple to use and understand.  Just try it.


FeedBack
~~~~~~~~
Please  send  donations,  bug  reports or a list of features you would
like to have included in future releases to:

Ultima Thule Software
```

Attn: Petter Nilsen
Mellomveien 138
N-9006 TROMSO
NORWAY

email: thor@thule.no
    or pettern@thule.no

MagicUserInterface
~~~~~~~~~~~~~~~~~~

This application uses

MUI - MagicUserInterface

(c) Copyright 1993 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXX.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 20.- or US$ 15.-

to

Stefan Stuntz
Eduard-Spranger-Strae 7
80935 Mnchen
GERMANY

History
~~~~~~~

930921 - Version 1.60 Beta 2:
- Should no longer crash at startup if MUI doesn't exist.
- Fixed problems with doing Stats if only the first element in the listview was selected

931004 - Version 1.63:
- Fixed even more startup troubles.

```
940101 - Version 3.16 (by Petter Nilsen)
- Cleaned up and ported to bbsread.library v3.
- A few visual enhancements.
```

ListEvents

This utility is located in the bin directory and is used to get information about the events outstanding without having to start THOR. The command template is

```
ListEvents [BBSNAME] [SHORT] [FULL] [DELAY] [USAGE]


BBSNAME - bbsname if  you  only  want  the  events  on one BBS
                  listed.   If left out, all BBSes will be listed with
                  their events.

SHORT - Will list information in a short form.

FULL - Will also list BBSes without events outstanding.

DELAY   - Will wait the given number of seconds before returning

USAGE - Will give a command template and other information.
```

PackBBS

This utility is located in the bin directory and is used to pack and purge conferences based on the values configured in THOR. The command template is

```
PackBBS [BBSNAME] [ALL] [QUIET] [DELAY] [USAGE]


BBSNAME - bbsname if you only want to pack the  conferences on
          one BBS.

ALL - Will pack all conferences on all BBSes.

QUIET - Will suppress all output from this command.

DELAY   - Will wait the given number of seconds before returning

USAGE - Will give a command template and other information.
```

Should be perfect for a cron-job...

WatchTHOR

```
------------------------------------------------------------------------
                              WatchTHOR

     Copyright  1995 Ultima Thule Software, All Rights Reserved.

                         Author: Petter Nilsen
------------------------------------------------------------------------

WatchTHOR  is a small notification program to be used with THOR 2.0 or
higher  to  notify  you  or start a specific program when new messages
arrive  in a conference.  It is pure and can be made resident.  It can
only  be run from shell.  WatchTHOR has no code to handle startup from
Workbench,  so  do NOT start it from Workbench!  It was never meant to
be  a  do-it-all program for notification, someone else can make that.
However,  since  several  users has requested such a possibility and I
had  2  hours of free time, I decided to make something.  Whenever the
developerkit  is  made  available,  I'll  make  the source code public
domain.  But until that happens, WatchTHOR is freely distributable but
NOT in the public domain.

<insert standard disclaimer here>

WatchTHOR accepts the following arguments:

SYSTEMNAME
~~~~~~~~~~
The name of the system to watch.

CHECKDELAY
~~~~~~~~~~
The number of minutes between each check.

Default: 5

CONFERENCE
~~~~~~~~~~
The name of the conference to check.

Default: EMail

RUNCOMMAND
~~~~~~~~~~
```

```
The command to run when new messages has arrived in a conference.
No requester will open.

Example:

1> watchthor "news&mail" checkdelay 5

This  command  line  will  check  for  new  messages  in EMail every 5
minutes,  and  bring  up  a requester on the frontmost screen with how
many new messages that arrived.

1> watchthor "news&mail" checkdelay 5 runcommand "dsound NewMail"

Will  do  as  the  example  above, except that it will run the program
dsound with "NewMail.iff" as argument.

- Petter Nilsen (pettern@icenet.no), October 29th 1995
```

ScriptServer/CfgType

```
**************************************************************************

950423 - Tech dokumentation for ScriptServ and CfgType

**************************************************************************
ScriptServ.doc

ScriptServ is used by the comm program script to obtain data from the
bbsread.library. The data obtained are the data about the events and the
setup for the bbs.

ScriptServ has the following parameters:

BBSNAME - Name of bbs to serve.

EVENT/N/K/M - The events the script can perform. Only events of the types
given is written to the events datafile called "T:bbsname.EVENT".

DONE/K - Number of event to mark as done.

ERROR/K - Number of event to mark with an error.

WRITECFG/S - Will write a file with configuration for the bbs. The file is
called "T:bbsname.CONFIG"

PACKAGEDONE/S - Mark all packed events as done. This should be used after a
package of events has successfulle been transfered. Example a QWK package.
```

```
RENAMEGRAB/S - Check for grabs to rename. Will give the grab a number so it
will be parsed in the correct order if there are more than one.

PRIVUPLOADFILE/K - Check if a comm program login capture from a ABBS/MBBS
BBS contain messages about private files.

UPDATEFILENOTES/K - Set the filenote of downloaded files to the short
description of the file. The description most have been catured from a
ABBS/MBBS BBS.


********** Format of configuration file:

VERSION                 3
BBSID                   <- ID of bbs.
BBSTYPE                 <- Type of bbs.
BBSDIR                  <- Directory where the files for this bbs is.
POSTCONF                <- (Assumed) name of conference for private mail.
UPLOADDIR               <- Uploaddirectory.
USERNAME                <- Username configured for this bbs.
USERSTREET              <- Userstreet configured for this bbs.
USERADDRESS             <- Useraddress configured for this bbs.
USERCOUNTRY             <- Usercountry configured for this bbs.
USERPHONE               <- User phone number configured for this bbs.
NEWFILES                (y/n)
AUTOPRIVDNL             (y/n)
AUTOLOGOFF              (y/n)
USECOLORS               (y/n)
ANSIMENUES              (y/n)
BULLETINS               (y/n)


********** Format of event data file:

EVENTNR                 <- Use when marking done/error
EVENT                   <- ID number for event
<List of tags in same order and number as in type config. One on each line.>
EVENTNR
EVENT
............
............
and so on.


********** Event ID numbers:

EVE_ENTERMSG        0  - Enter message
EVE_REPLYMSG        1  - Reply message
EVE_JOINCONF        2  - Join conference
EVE_RESIGNCONF      3  - Resign conference
```

```
EVE_DOWNLOAD        4  - Download file
EVE_UPLOAD          5  - Upload file
EVE_CONFLIST        6  - Get conflist
EVE_CONFIGUREBBS    7  - Set correct configuration on the bbs.
EVE_SENDUSERINFO    8  - Send userinfo
EVE_FORWARDMSG      9  - Forward a specific message to another user
EVE_GETNEWFILES    10  - Get new files listing
EVE_DOCOMMAND      11  - Command the script should do
EVE_AUTOLOGOFF     12  - Autologoff on next session
EVE_RETRACTMSG     13  - Retract message from bbs
EVE_RETRACTFILE    14  - Retract file from bbs


********** Event tags id numbers:

EV_ToName              0
EV_ToAddr              1
EV_Subject             2
EV_Conference          3
EV_RefNr               4
EV_RefOrginalNr        5
EV_RefId               6
EV_MsgFile             7
EV_Private             8
EV_LocalFile           9
EV_Directory          10
EV_FileName           11
EV_DownloadNotify     12
EV_FileDescr          13
EV_DetailedFileDescr  14
EV_FromMessageNr      15
EV_ToMessageNr        16
EV_CommandString      17
EV_Boolean            18
EV_Date               19
EV_PGPSignID          20
EV_PGPEncryptID       21
EV_RefConference      22
EV_Urgent             23
EV_Important          24
EV_Confidential       25
EV_ReturnReceipt      26
EV_Encode8bit         27

The different event tags are described in BBSReadRexx.doc/WRITEBREVENT.

-------------------------------------------------------------------------
******** Example NComm script:
; $VER: example.script 3.1 (23.04.95)

REQUEST OFF
```

```
set $done = "FALSE"

; Get configuration and events for the bbs. $BBSName is defined in the
; startup script.
cli $THORPath"bin/ScriptServ \""$BBSName"\" WRITECFG EV 0 EV 1 EV 2 EV 3"

; Open configfile and read the configuration
varfile "t:"$BBSName".CONFIG"

readvar $CfgVer
readvar $BBSID
readvar $BBSType
readvar $Path
readvar $PostConf
readvar $UpLoadDir
readvar $UserName
readvar $UserStreet
readvar $UserAddress
readvar $UserCountry
readvar $UserPhone
readvar $NewFiles
readvar $AutoPDnl
readvar $AutoLogoff
readvar $UseColors
readvar $AnsiMenues
readvar $Bulletins
varfile close

; Check for correct version
if !$CfgVer == "3" then message "Incompatible version of ScriptServ"
if !$CfgVer == "3" then goto AllDone

cli "delete >nil: \"t:"$BBSName".CONFIG\""

[...] ; Logon handling here.


;***************** Event loop


; There will be no eventfile if there are no events.
if !exists "t:"$BBSName".EVENTS" then goto afterEvents
varfile "t:"$BBSName".EVENTS"

eventLoop:
set $done = "FALSE"

readvar $eventnr
readvar $event
```

```
; Loop until the end of the file is reached.
if $eventnr == "EOF" then goto evLoopEnd
if $event   == "EOF" then goto evLoopEnd

if $event == "0"  then gosub EnterMsg
if $event == "1"  then gosub ReplyMsg
if $event == "2"  then gosub JoinConf
if $event == "3"  then gosub ResignConf

if  $done == "TRUE" then cli "run >nil: "$THORPath"bin/ScriptServ \""$BBSName"\" DONE "$e
if !$done == "TRUE" then cli "run >nil: "$THORPath"bin/ScriptServ \""$BBSName"\" ERROR "$

goto eventLoop

evLoopEnd:
varfile close

afterEvents:

[.....] ; Rest of script and subroutines follow.

end

;*******
EnterMsg:
; Read event tags for enter message event.
readvar $conf
readvar $sendto
readvar $msgfile
readvar $subject
readvar $private
if $private == "" then set $private = "0"
if !$private == "0" then set $private = "y"
if $private == "0" then set $private = "n"


[....] ; Perform enter message event.

set $done = "TRUE"
return

; For more examples check out AMBBS.NComm.script

------------------------------------------------------------------------
************************************************************************
CfgType.doc

CfgType is used to make the configuration for a BBStype. It has the
following parameters:
TYPENAME              - Name of type to configure.
NEWTYPENAME/K         - New name of type is the type should be renamed. Only
```

```
                          unused types can be renamed.
MSGPARSER=MP/K          - Command to parse and add the messages recieved to the
                          database.
LINELEN=LL/K/N          - Max length of message lines the BBSType allows.
SUBJECTLEN=SL/K/N       - Max length of subject the BBSType allows.
FILEDESCRLEN=FL/K/N     - Max length of filedescrition.
CHARSET/K               - Which charset the bbs uses: Available charsets are
                          ISO/IBN.
AVAILSCRFLAGS=ASF/K     - Which scriptflags are available for this BBSType.
                          Hex-number with the flags or'ed.
FILENAMELEN=NL/K/N      - Max length of filenames on this BBSType.
EVENTPACKER=EP/K        - Command to pack events into a package to transfer.
NOEVENTPACKER=NEP/S     - This BBSType has no eventpacker.
EVENTARCHIVER=EA/K      - Name of archivertype to use for achiving the
                          eventpackage. The archiver must have been configured
                          in the global configuration.
NOEVENTARCHIVER=NEA/S - This BBSType should use no archiver for the packed
                          events.
CONFIGEVENT=CFGE/M/K  - Config tags for the events. See below for docs.
DELETETYPE/S            - Delete configuration for this type.
DELETEEVENT/K/N         - Delete tag configuration for a event.
SETTYPEFLAGS=STF/K      - Set type flags for this bbstype.
CLEARTYPEFLAGS=CTF/K  - Clear type flags.
ACCEPTPATTERN=AP/K      - Pattern to match grabs with. Default is #?.
INITMSGFILE/K           - Command to initialize message files.
EXTCONFIG/K             - Command for extrernal bbs configuration.
QUOTETYPE               - Default quote type for this bbs.

A eventdescr consist of: eventnr:needtag,needtag,...:opttag,opttag,...
Examples:
cfgtype MBBS ConfigEvent 0:0,1,2,4:5 ConfigEvent 1:1,2,3,4:5
  Configures 2 evenets for the bbstype called MBBS. These are:
  - EVE_ENTERMSG (0) which needs the following tags:
      ET_SendTo (0), ET_Conference (1), ET_MsgFile (2), ET_Subject (4)
      Optional tags are: ET_Private (5)
  - EVE_REPLYMSG (1) which needs the following tags:
      ET_SendTo (0), ET_Conference (1), ET_MsgFile (2), ET_ReplyTo (3)
      ET_Subject (4)
      Optional tags are: ET_Private (5)

The definitions of each tag see ScriptServ.doc

For examples of the use of CfgType see Thor:s/cfg#? files.

If you have any questions concerning ScriptServ or bbs type configuration,
feel free to send an email.
```

BaseManager

940912 BaseManager V3.55

for use with BBSRead Release 2.0.

Copyright 1994 Ultima Thule Software, All Rights Reserved.
Author: Eivind Nordseth

BaseManager is the tool to use when you experience trouble with your database. It also contains routines to update older revisions of the database.

BaseManager is located in the <Thor:>bin/ directory, and it has the following parameters/flags: VALIDATE, SALVAGE and CONVERT.

VALIDATE
~~~~~~~~

Sets the validate flag for a bbs. To set the validate flag for all bbses use "ALL" as bbs name. Use this when you have trouble adding events or messages.

SALVAGE (NOT IMPLEMENTED YET FOR V3)
~~~~~~~

CONVERT
~~~~~~~

Convert older versions of the bbsread.library database to the format used by bbsread.library V3. All not-deleted information in the database will be converted. Normaly will all messages have the same message number in the converted database as in the old one. The same counts for events to. Only users with current messages to or from them will be added to the new user database.

BaseManager is normaly executed with the CONVERT switch by the Installer script. The convertion of the database can be done /by/ /hand/ by by executing BaseManager with the CONVERT switch.

After the convertion must you update the bbstype configuration. This is done by executing the cfg#? scripts in the s-directory in Thor's home directory.

Since the converted database takes some more storage place (due to speed optimized storage) you must be sure to have *at* *least* 0.5 of the database size as free space available on the disk. (The storage space used can later be reduzed by using the xpk packing option.)

Remeber to take a backup before using the CONVERT switch.

HINTS & TIPS
~~~~~~~

```
~~~~~~~~~~~~
*Always* take a backup of the data before using BaseManager with the
SALVAGE or CONVERT switch..
```

15 Character sets supported by THOR

THOR handles messages in many different character sets. Incoming messages are converted from the configured charset to the standard Amiga charset (ISO). Outgoing messages are converted from ISO to the configured charset.

Supported character sets:

ISO - ISO Latin 1 charset.
IBN - IBM charset used in Norway. Same as IBM CodePage 865.
SF7 - 7-bit charset. Sweden and Finland.
NO7 - 7-bit charset. Norway.
DE7 - 7-bit charset. Denmark.
US7 - 7-bit charset. USA.
UK7 - 7-bit charset. United Kingdom.
IT7 - 7-bit charset. Italia.
PO7 - 7-bit charset. Portugal.
SP7 - 7-bit charset. Espania.
GE7 - 7-bit charset. Germany.
FR7 - 7-bit charset. France.
IBM - IBM charset. Same as IBM CodePage 437.
IBMM - IBM multinational charset. Same as IBM CodePage 850.
APPLE - Apple charset.

THOR does not support character sets with more than 256 characters.

16 Trouble-shooting guide

`Question`

What is this "system" you talk about all the time?

`Answer`

To put it simple, a system is either a particular BBS, an Internet provider or a Fido site. You can use THOR to read messages or postings from many systems.

`Question:`

How do I choose which packet to open, and why is there no file requester?

`Answer:`

You don't need a file requester. Instead of manually choosing which packet to open, THOR checks your download directory for packets with names matching the offset

(or pattern) you have defined for the system in the Systems Configuration part of the ConfigTHOR program.

`Question:`

I get the following error message when I start THOR: "Failed to load the configuration datafiles."

What's wrong?

`Answer:`

1. Since the message database is located on disk, it is sensitive to whatever might happen to the disk. In case of a disk crash or similar problem, use one of the commercial or shareware tools that can fix such problems. DiskSalv, Quarterback Tools or AmiBack Tools are all recommended for this purpose.

2. One or more of the datafiles that THOR needs might be corrupt or damaged. Take a backup of your database, then use [BaseManager], page 66 to fix the problem. If not even that helps, you may have to delete the system and make a new one.

See [Repairing a damaged database], page 71.

`Question:`

I have configured a system with one of the QWK types, but when I exit the system after having written a message, I get an error message stating that a file called "control.dat" can't be found. What's wrong?

`Answer:`

PackQWK needs the file "control.dat" before a package with messages can be packed. The solution is to call the system in question and collect a message packet which must be parsed into the THOR database. In this packet, the file "control.dat" is included and THOR/PackQWK should not give this error message anymore.

`Question:`

When I try to configure a new system, the "Type" gadget is empty. Why?

`Answer:`

THOR isn't installed correctly. Re-run the installation, reboot and try again. If you don't want to install again, you might execute the cfg* scripts i <YourThorDir>s/ from shell with the command "execute".

`Question:`

When I quit THOR, it doesn't free all the memory used. Why?

`Answer:`

bbsread.library uses a buffering system. The buffers will not be freed immediately by the library. The library will flush the buffers after 5 minutes if the library isn't used anymore. You can flush the library manually with the "avail flush" command, but AmigaOS will automatically flush the library from memory when the memory is needed by some other program if the library is not in use anymore.

`Question:`

The ABBS/MBBS NComm script do not seem to work at all on MBBS systems, and the script stops on the first prompt after it has written my name and password. What is wrong?

`Answer:`

Make sure you have selected "English" as language on the system.

`Question:`

THOR does not find programs I start from within THOR, like the ThorStat program, even if the directory these programs reside in is in my path. I have made sure I have the correct path statement in my user-startup. What it wrong?

`Answer:`

If you have made sure you use the ADD keyword to the path command in your user-startup and that this directory actually exists, one possible reason might be that you start THOR as a WB Prosess from ToolManager. The WB Startup code in ToolManager is buggy and ignores any path, so you should start THOR as a CLI prosess from ToolManager instead (WBStart-handler 1.4 included with ToolManager 2.1a fixes this problem). THOR will inherit the path even if started from Workbench since THOR converts itself to a cli prosess on startup.

ParM will show similar (bad) behaviour when started from WB, so start if from CLI whenever possible.

`Question:`

ARexx scripts won't work correctly and says they can't find "WaitForPort".

`Answer:`

See the previous answer.

`Question:`

THOR doesn't seem to work correctly while running on a MultiUser partition, what is the problem?

`Answer:`

Due to a bug in MultiUsers handling of async lowlevel dos packets, the files in all THOR's datadirs must be set up without any user or group. This can be done with the following command:

mprotect 'getenv thor/bbsdatapath' RWD group RWD other RWD all

`Question:`

THOR has problems unarchiving and parsing my message packets. What can be wrong?

`Answer:`

1. Check that you do not have any extension on the MailPacket name in Systems Configuration. If a QWK packet is names "somebbs12.qwk", the correct mailpacket name will be "somebbs".

2. Check that you have the unarchiver (Unzip, Lha, etc.) in the path. If you run THOR from ToolManager, make sure you start THOR as a CLI prosess and NOT as a WB prosess.

3. Check the the MailPacket actually is in the format you have configured the system in THOR to accept.

4. Check that you haven't configured a relative path to your download directory. Use the complete path or assign to this directory.

`Question:`

I have moved my database to some other directory, but THOR will not find my database in the new location. What's wrong?

`Answer:`

See Chapter 17 [Moving the installed database], page 72 for a description on how to do this correctly.

`Question:`

Sometimes THOR will not remove the waitpointer even when finished doing prosessing of data. THOR will still recognize mouse clicks etc.

`Answer:`

This can happen with certain commodities that change the mouse pointer in various ways. Particular older programs (eg. QMouse) is known to not handle this correctly. Try running without these programs.

`Question:`

THOR seems to be slow in adding messages to its database, what can be done to speed it up a bit?

`Answer:`

- Turn off the "Add Users" option on the system in question in Systems Config -> Import/Export.

- Open the user database window and delete all the users you do not want in the database. Then select Maintenance in this same window.

- Set up more buffers and higher buffer size in Global Config -> Database options. When using the TCP module, 2-3 buffers for each newsgroup you subscribe to is a good setting.

- If you use XPK packing, turn off "Auto XPK packing" in Global Config -> Database options.

Repairing a damaged database

Starting from THOR 2.4, a script for repairing the database has been included. Look for "ValidateThor" in the THOR bin/ directory. Note the following about this procedure:

- Only run ValidateThor after a reboot.
- For the best stability leave the system alone while validating.
- Be aware that validating the databases can take a long time.
- ValidateThor will perform a pack/purge after deleting corrupt messages.
- Use Sushi to avoid error requesters. Sushi can be found on AmiNet.

ValidateThor takes a single argument, namely the name of the system in THOR to repair. Example:

ValidateThor "News&Mail"

ALL can be used instead of the system to validate all systems in THOR.

Afterwards, you might consider doing the following steps:

- Start Thor, enter the system and set the view ALL mode. Go trough all conferences to see if there are any messages in the messagelist window that are marked with *BAD* *DATA* (These may show up on the last messages in the list.). Multiselect (with the shift key) all messages with "*BAD DATA*" or "Failed to get message data" and then select Delete msgs from the menu.

If this do not help, try to delete the offending messages by hand and pack/purge the conference.

17 How to move a THOR installation

Sometimes it might be useful to move the THOR installation to another disk or somewhere else on the same disk. This can be done in the following fashion:

1. Make sure you are not running THOR and flush the memory with the command "avail flush" from a shell.

2. Move the database to whereever you want it.

3. Change envarc:THOR/THORPath to point to the new directory you have THOR in. If you intend to use the included NComm scripts, you MUST make sure there's a newline in this path statement due to a bug in NComm.

4. Change envarc:THOR/BBSDataPath to point to the new directory you have you database. The database is usually stored in a "data"-subdir under the THOR directory.

5. Reboot.

18 What the magazines say about Thor

"Thor gets TCP capability - The execellent Norwegian multipurpose electronic mail browser, Thor, has had a support package released that allows it to interact directly with dial-up news and E-Mail. This turns Thor 2.0 into one of the most powerful complete electronic mail solutions around."
- CU Amiga, September 1995, page 112

"Never has it been easier to read email and news on the Amiga. The program that makes it so easy is called Thor, but it is not a new program. It's an offline reader for various BBS formats that has grown and become the best email/news system we have seen on the Amiga so far." [...] "The program has now got a very nice TCP-module for handling email and news. In many ways it beats everything I've seen on both MAC and PC." (translated from Swedish)
- DatorMagazin, September 1995, page 50-51

"Thor 2.1 released - The shareware Thor package provides a superb interface for mail and news handling and has been steadily improved since its incarnation. With the advert of the newly released version 2.1, many improvements have been added not least the inclusion of the ThorTCP package. This provides dedicated Internet Mail and News slurping clients (NNTP, SMTP and POP3) for Thor which turns it into an entire solution providing not only the means to read and write the electronic mail/news but also to up/download it from your provider. Many bugs fixed and small improvements have been added to the software which surely make it the No.1 integrated mail and news package for the

Amiga."
- CU Amiga, November 1995, page 106

"Thor is really the only choice for email and news handling that works really seamlessly. One can only be more and more impressed by this program." (translated from Swedish)
- DatorMagazin, November 1995, page 50-51

19 Thanks

THOR has been a team work from the start of this project in late 1992. Many people have contributed to the project in many ways.

The authors of THOR wish to send thanks to the following:

Jacob Thyness - for the great THOR logo on the disk labels, on our Web-site and in the Startup window!

Troels Walsted Hansen
- for many of the useful ARexx scripts, the ARexx documentation, and for beta testing

Harald Kuhr - for the normal THOR program icon.

Matthias Bartosik
- For the German locale and for beta testing.

Peter Larsen - For making a lot of suggestions and sending us QWK packets that THOR had problems parsing.

Rune Hansen - for drawing the hammer originally used on the disk labels.

Roman Patzner - for the MagicWB icon for THOR

Phil Vedovatti (vedovatt@u.washington.edu)
- for the NewIcons for THOR, ConnectTHOR, ConfigTHOR etc.

Johnny Bonde - Danish locale
Henrik Dissing - Danish locale and beta testing
Krister Larsson - Swedish locale
Jean-Michel Bezeau - French locale
Claudio Alviggi and Francesco Dipietromaria - Italian locale
Ralf Bosz - Dutch locale
Remco van Hooff - Dutch locale and beta testing
Andreas E. Bombe - German locale

ATO translations (coordinated by Thomas Andersson <steiner@alfaskop.net>):

Language: Catalan (catal)
Translator: Lloreng Grau <llg@cryogen.com>
Proofreader: Lloreng Grau <llg@cryogen.com>

Language: Finnish (suomi)
Translator: Mika Lundell <c71829@uwasa.fi>
Proofreader: Linus Silvander <linus.silvander@parnet.fi>

Language: Serbian (srpski)
Translator: Andrija Antonijevic <TheAntony@bigfoot.com>
Proofreader: Ljubomir Jankovic <lurch@afrodita.rcub.bg.ac.yu>

Language: Spanish (espaol)
Translator: Ignacio Garca <igarcia@readysoft.es>
Proofreader: Samuel A. Aguilera <slyguy@mx2.redestb.es>

Language: Portugse (portugus)
Translator: Fernando Nunes <fnunes@individual.eunet.pt>
Proofreader: Alexandre Gabriel <chryse@mail.telepac.pt>

Additional Beta testers, some are still beta testers, some are not:

Gary Peake
Shaun Downend
Bob Carter
Neil Bothwick
John S. Burger
Jonathan Rhys Evans
Grahame Fendle
Torge Hjorth
Stein Bjrndal
Ola Paulsen
Jrn Innset
Tommy Larsen
Lars Dybfest
Eirik Synnes
Vebjrn Forsmo
Andreas Dobloug
Christopher Naas
Christian Sandberg
Nils Sudmann

....and last but not least, the hundreds of registered users of THOR that has supported us with registrations, kind words, scripts and comments.

THANKS!

20 Changes to this documentation

This section lists changes that have been made to this documentation during the last few THOR releases. It is not a list of changes to THOR itself, see the THOR.history file for these.

`Changes in this doc for THOR 2.2:`

- Added a separate chapter on how to repair the message database should this be needed. See [Repairing a damaged database], page 71.

- Updated the docs on the information shown in the userdatabase main window. See [User database], page 17.

- Updated to docs on the menus to include the new options. See Chapter 6 [Menus], page 24.

- Updated to docs on the [Group by], page 12 option to include the new Subject+Thread group method.

- Updated the docs on [Join], page 11 conference.

- Updated the docs on the [Messagelist Window], page 13 to include information about the new multiselection feature.

- Added docs on the new options in the [Reply], page 54 and Enter message window.

- Added docs on the maintenance option in the kill/emphasize database. See [Kill database], page 18.

- Added docs on the new [WatchTHOR], page 60 program.

- Added docs on the new shortcut for turning on/off the extra header fields. See [Shortcuts], page 50.

- Added docs on the SHIFT-ESC shortcut combination. See [Shortcuts], page 50.

- Added docs on the 2 new [Group by], page 12 modes Subject Alpha and Subject+Thread Alpha.

- Added docs on the new options in the newsbrowser in ConnectTHOR. See [ConnectTHOR], page 36.

- The Show Headers and Show Styles buttons present in the mail window is documented in [Message header], page 53.

- The various MIME decoding/encoding methods supported is documented in the Chapter 12 [MIME support], page 55 section.

`Changes in this doc for THOR 2.3:`

- Kjell and Magne updated the MIME section and the about section.

- Added the keypad shortcuts to the [Shortcuts], page 50 section.

- Updated the [Group by], page 12 section.

- Updated the docs on Style Tags in the [Visual Options Window], page 6 section.

- Updated the docs on the [Messagelist Window], page 13 to describe the new gadget.

- Updated the ConfigTHOR.guide in the reply string section and added the 2 new forward codes.

- Updated the [Mark Window], page 23 on the new "mark login conference" function.

- Added the new Catchup Upto to the Chapter 6 [Menus], page 24 session.

- Did not find a place for the env variable "THOR/SYSTEM_ENTRY_CONFLIST" in the docs, read THOR.history for more info about it.

- Added docs on "Auto Show First Message", "Delayed Msg Unmark" and the new external editor command line formatting code in ConfigTHOR.guide.

- Added docs on the new [Pack/Purge Window], page 8.

- Updated the docs on the [CfgTCP], page 38, [ConnectTHOR], page 36, [SendTCP], page 40 and [GetTCP], page 41 programs.

- Updated the docs for THOR 2.4.

`Changes in this doc for THOR 2.5:`

- Added docs on the changes in [Pack/Purge Window], page 8.

- Added docs on the changes in the [Kill database], page 18, including a few tips and clarifications.

- Updated Chapter 12 [MIME support], page 55 to include information about external icons.

- Added information about double-clicking on names in the [User database], page 17.

- Updated the [ConnectTHOR], page 36 documentation to include the 2 new arguments and the new "Filter crosspostings" option in the browser.

- Updated the [Group by], page 12 section to include info that it can be used even in All and Search mode now.

- Updated the [Visual Options Window], page 6 documentation with the new link features along with the option to show pictures automatically.

- Updated the documentation on the forward code for first and last name, that was implemented some time ago but never documented.

Index

$

$time ... 24
$ver ... 24

A

ABBS ... 29
About ... 1
Add conference ... 11
Add Macro ... 24
Address ... 17
Alias ... 17
All Confs ... 21
All First grouping ... 12
AmiTCP ... 30
ANSI Support ... 6
Areas ... 15
ARexx in the FSE ... 47
Arguments ... 5
Author ... 12
Authors ... 1
Autologoff ... 24

B

BaseManager ... 66, 71
BBBS ... 29
Beta-testers ... 73
BlueWave ... 30

C

Cancel ... 18, 21
Capabilities in THOR ... 3
Catchup ... 12
CfgTCP ... 35, 38
Changes to this documentation ... 75
Character sets ... 68
Clear ... 15
Clear/Mark ... 15
Color Quote ... 6
Comment ... 17
Communication ... 42
Conferences ... 14, 18, 21
Configuration ... 5
Configuring TCP ... 35, 38
Connecting to the Internet ... 36
ConnectTCP ... 36
ConnectTHOR ... 35
Contact Address ... 1

D

Date ... 15
Delete ... 15, 17, 18
Delete Dups ... 24
DeleteMsg On/Off ... 24
Detailed File Descripion ... 11
Disclaimers ... 2
Dock Window ... 10
Documentation changes ... 75
Double Quote ... 6
Download ... 10, 15
Download notification ... 11
Drop conference ... 11

E

Edit ... 18
Edit Macro ... 24
Editor ... 43
Editor DOS scripts ... 47
EMail ... 35
Emphasize Database ... 18
Emphasize Type ... 18
End Msgnr ... 21
Enter ... 54
Eventlist ... 21
Events ... 13
Execute Macro ... 24
External utilities ... 57

F

F-keys ... 24
Features in THOR ... 3
FIDO_EXPORT ... 31
FidoNet ... 31
File Database ... 15
File Description ... 11
File search ... 15
Files ... 15
Find and mark messages to you as unread ... 23
First time startup ... 4
Followup ... 54
Followup to a message ... 54
FORCETOFRONT ... 5
From user ... 21
From-Address ... 18
From-Name ... 18
FSE ... 43
FSE ARexx ... 47
FSE DOS scripts ... 47
Full Screen Editor ... 43
Function Dock ... 10

G

Get Conference List ... 10
Get Current Msg ... 18
Get Newfiles ... 24
GetTCP ... 35, 41

Getting started with the TCP modules 35
Group messages 12
Grouped 12
Grouped alpha 12
Grouped author 12

H

Haze 18
Haze Levels 18
Header 53
HELP 68
Hippo 29
Hold 15
Hotkeys 50
How to install THOR 4
How to register THOR 2

I

ICON_X 5
ICON_Y 5
Iconify 24
Important 18
Installation 4
Internal Editor 43
Invert 15

J

Join conference 11

K

Keep 18
KeepMsg On/Off 24
Kept 21
Kill Database 18
Kill Type 18

L

Leave BBS 24
List of messages 13
ListEvents 59

M

Magazin reviews 72
Mail 35
Main Window 9
Maintenance 15, 17
Mark a group of messages as unread 23
Mark current message as unread 23
Mark message with given number as unread 23
Mark messages from given date as unread 23
Mark messages from login state as unread 23
Mark messages in this thread as read 23
Mark messages in this thread as unread 23
Mark messages with this subject as read 23
Mark msgs from conf. login state as unread 23
Mark unread msgs from current author as read 23
Mark unread msgs not from or to you as read 23
Marking messages 23
MBBS 29
Menus 24
Menus in the FSE 44
Message Formats 29
Message grouping 12
Message list 13
Message reading 49
Message search 21
Message window 9
MIME 55
Mine First grouping 12
Moving the installed database 72
Msg Text 18
MsgNr grouping 12
MsgText+Subject 21
Multipurpose Internet Mail Extension (MIME) 55

N

Name 17
NComm 43
NComm Scripts 42
NetNews 35
New 17, 18
Newfiles Window 10
News 35
NewsBrowser 35
NNTP 35
None 21
NOPARSEGRAB 5

O

Omen 30
Open 9
Other communication programs 43

P

Pack Conf 8
Pack/Purge Window 8
PackBBS 59
Packing messages 8
POP3 35
Print Message 24
Problems 68
Public Screen 24
PUBSCREEN 5
Purge Conf 8
Purging messagebase 8

Q

Quit 24
Quotes 72
QWK 29

R

Read modes 12
Reading messages 49
Reference grouping 12
References 18
Registration 2
Remove 15
Repairing a damaged database 71
Reply 54
Replying to a message 54
Rescan 9
Resign conference 11
ROT-13 24

S

Save 15, 17, 18
Save Macros 24
Save Message 24
SCREENPICTURE 5
Script Command 24
Scripts 42
ScriptServer 42, 61
Search 15, 21, 24
Search Window 21
Send Userinfo 12
SendTCP 35, 40
Set 'last message read' pointer 23
Shareware 2
Shortcuts 49, 50
Shortcuts in the FSE 44
Show ALL 12
Show headers button 53
Show ScriptMsg 24
Show Search 12
Show style tags button 53
Show Unread 12
Signature system 24
SMTP 35
Sort 15, 17
SOUP 30
Start Msgnr 21
Starting THOR 4
Startup Window 9
String 21
Style Tags 6
Subject 18, 21
Subject alpha grouping 12
Subject grouping 12
Subscribe newsgroup 11
SYSTEM 5
System Filename 11
System types 29

T

TCP_ONLINE 35, 42
Thanks 73
ThorStat 57
Threading 49, 50
To user 21
To-Address 18
To-Name 18
Tooltypes 5
Trouble-shooting guide 68
Twit list 18
Type 18

U

Unix 30
Unsubscribe newsgroup 11
Upload 11
uqwk 30
UQWK_SOUP 30
Urgent 18
User Database 17
UUCP 33

V

Visual Options 6

W

WatchTHOR 60
WINDOWPICTURE 5
Write 54
Writing a new message 54
Writing scripts 61

X

XPK packing 8
XPOS 5

Y

YPOS 5

Short Contents

. 1
1 About 1
Disclaimers 2
2 Registration 2
Features in THOR 3
3 Installation 4
Starting THOR for the first time 4
Configuration 5
THOR Tooltypes and arguments 5
4 Windows in Thor 6
Visual configuration window 6
Packing and purging window 8
Startup window 9
Main reader window 9
Newfiles window 10
Message overview/list window 13
Conference overview window 14
File database window 15
User database window 17
Kill/emphasize window 18
Event window 21
Search window 21
Mark window 23
5 Signature system 24
6 Menus 24
7 System types and formats supported 29
The ABBS/MBBS system type 29
The QWK system type 29
The Hippo system type 29
The BlueWave system type 30
The Omen system type 30
The SOUP system type 30
The FIDO system type 31
The UUCP system type 33
The TCP system type 35
Getting started with the TCP Internet modules 35
The ConnectTHOR Internet module 36
The Config TCP Internet configuration program 38
The SendTCP Internet module 40
The GetTCP Internet module 41
The TCP_ONLINE Internet system type 42
8 Scripts and communication 42
Using NComm with the BBS modules 43
Using other communication programs with the BBS modules 43
9 THOR Full Screen Editor 43
Shortcuts in the FSE 44
10 FSE Menus 44
The FSE Project menu 44

The FSE Macro menu 45
The FSE Miscellaneous menu 45
The FSE ARexx menu 46
FSE DOS scripts 47
FSE ARexx 47
11 Message reading and writing 49
Shortcuts when reading messages 50
The message header 53
Reply to a message 54
12 MIME support in THOR 55
13 Features for the Power User 56
14 External utilities 57
ThorStat 57
ListEvents 59
PackBBS 59
WatchTHOR 60
ScriptServer/CfgType 61
BaseManager 66
15 Character sets supported by THOR 68
16 Trouble-shooting guide 68
Repairing a damaged database 71
17 How to move a THOR installation 72
18 What the magazines say about Thor 72
19 Thanks 73
20 Changes to this documentation 75
Index 77

Table of Contents

. 1

1 About 1

Disclaimers 2

2 Registration 2

Features in THOR 3

3 Installation 4

Starting THOR for the first time 4

Configuration 5

THOR Tooltypes and arguments 5

4 Windows in Thor 6

Visual configuration window 6

Packing and purging window 8

Startup window 9

Main reader window 9

Newfiles window 10

Function dock window 10
Getting a list of conferences/newsgrops 10
Download a file 10
Upload a file 11
Resign/unsubscribe a conference/newsgroup 11
Join/Subscribe newsgroup 11
Send Userinfo 12
Catchup unread messages 12
Show Unread/All/Search result 12
Grouping of messages 12
Open the events window 13

Message overview/list window 13

Conference overview window 14

File database window 15
User database window 17
Kill/emphasize window 18
Event window 21
Search window 21
Mark window 23
5 Signature system 24
6 Menus 24
7 System types and formats supported 29
The ABBS/MBBS system type 29
The QWK system type 29
The Hippo system type 29
The BlueWave system type 30
The Omen system type 30
The SOUP system type 30
The FIDO system type 31
The UUCP system type 33
The TCP system type 35
Getting started with the TCP Internet modules 35
The ConnectTHOR Internet module 36
The Config TCP Internet configuration program 38
The SendTCP Internet module 40
The GetTCP Internet module 41
The TCP_ONLINE Internet system type 42

8 Scripts and communication 42

Using NComm with the BBS modules 43

Using other communication programs with the BBS modules 43

9 THOR Full Screen Editor 43

Shortcuts in the FSE 44

10 FSE Menus 44

The FSE Project menu 44

The FSE Macro menu 45

The FSE Miscellaneous menu 45

The FSE ARexx menu 46

FSE DOS scripts 47

FSE ARexx 47

11 Message reading and writing 49

Shortcuts when reading messages 50

The message header 53

Reply to a message 54

12 MIME support in THOR 55

13 Features for the Power User 56

14 External utilities 57

ThorStat 57

ListEvents 59

PackBBS 59

WatchTHOR 60

ScriptServer/CfgType 61

BaseManager 66

15 Character sets supported by THOR 68

16 Trouble-shooting guide 68

Repairing a damaged database 71

17 How to move a THOR installation 72

18 What the magazines say about Thor 72

19 Thanks 73

20 Changes to this documentation 75

Index 77