Zangband

Bablos <bablos@cableinet.co.uk>

COLLABORATORS

	TITLE : Zangband		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Bablos <bab-los@cableinet.co.uk>	January 19, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Zangband

1.1 Zangband Documentation

```
            Zangband 2.2.6c
            ~~~~~~~~~~~~~~~

            Introduction
            Requirements
            Installation

            Settings
            Inscriptions
            Macros
            The Borg

            Variants
            Links

            Credits
            Author

      Written by Bablos, 05.09.99
```

1.2 Introduction...

```
    Introduction
    ~~~~~~~~~~~~

    Angband
    ~~~~~~~
    Angband is a single player role playing game. You choose a race and a
    class, then roll up statistics to describe your various abilities. Then
    you're unleashed upon the dungeons of Angband, ultimately dropping to
    dungeon level 100 and trying your damndest to kill Morgoth, Lord of
    Darkness.

    Along the way you encounter a myriad of creatures and uniques to hinder
    your progress, and objects and artifacts to help you achieve your goals.
```

Angband is loosely based on the works of JRR Tolkien, and had the game
Rogue as an ancestor, hence the term 'roguelike' that is encountered in
this guide and many other documents relating to Angband and it's variants.

Angband is constantly under development, and has been since it came out
in its most basic form back in 1985.

Zangband
~~~~~~~~
Zangband is an Angband variant. This is probably the most radically
different variant available, and in my opinion is the best around, even
beating plain old vanilla. The Z is for Zelazny, as Zangband is based
around Roger Zelazny's 'Amber' universe. The game has a revolutionary new
magic system derived from the game 'Master of Magic'. Other than this, the
game now has 26 race types, each having their own special abilities as
they reach certain levels, loads more monsters, a HUGE array of vaults of
all sizes, random artifacts in addition to the traditional set, new
objects, new player and object flags, and a new goal at the end of the
game.

For full details of all the changes made to Zangband throughout it's
history, please consult the z_update.txt file, either via the link in
the guide or by perusing the file itself which you will find in the
game directory.

1.3   Requirements...

Requirements
~~~~~~~~~~~~

Absolute Minimum
~~~~~~~~~~~~~~~~
You'll need at least this to run plain Zangband.

o Kickstart 1.3
o 1.4 Mb of memory
o 68000 – a faster processor is nice, but by no means necessary.

To run a borg compiled executable, you'll need at least 2Mb of free
memory, and a fast processor would definitely be a bonus too.

Recommended Level
~~~~~~~~~~~~~~~~~
This is what is recommend to get the most out of Angband:

o Kickstart 3.0 or better
o 2Mb of memory
o 68020 or better.
o A Hard drive.

Running Angband and it's variants under kickstart 1.3 is somewhat
restricted. Mark doesn't plan to support 1.3 further, and the assumption
will be made that everyone is using 2.04 or better. Don't worry though,
the game will still work under 1.3, it'll just do so without the bells,

whistles and shiny bits that Mark's put together for those of us that
aren't living in the stone age...

Machines that Zangband have been tested on;

o   A1200T, 68040/25, 603e/200, 48Mb FAST, 24xCDROM, 2.1Gb HD (Mark)
o   A1200, 68060/50, 603e/200, 32Mb FAST, 5.04Gb HD (Bablos)

1.4 Installing Zangband

Installing Zangband
~~~~~~~~~~~~~~~~~~~~

Installing to Hard disc
~~~~~~~~~~~~~~~~~~~~~~~~~
Couldn't be easier – just unpack the archive to the directory of your
choice! Then it all depends on what kickstart your machine has;

o KS 2.0+
  If you're lucky enough to have a kickstart 2.0 or better machine, then
  all you need to do to get Zangband up and running is to run the program
  Zangband.

o KS 1.3
  If you only have Kickstart 1.3, you'll also need to place a few
  commands into your S:startup-sequence file, or make a script file if
  you're feeling particularly adventurous and execute it as and when you
  intend to play Zangband. Anyway, the commands required are;

  assign Zangband: DH1:Games/Zangband
  path add Zangband:

  Obviously, replace the DH1:Games/Zangband with the path to the Zangband
  directory on your system. Make sure that this new Zangband: assignment
  points to the right place! It should point to the main directory
  containing all the file, data, apex, raw etc. directories.

Floppy installation
~~~~~~~~~~~~~~~~~~~~~
What, no hard drive? Never mind. You can just about install Zangband
onto floppy discs, but it's a tight fit. Your friendly neighbourhood
compression programs might help you out here.

First get two empty, formatted discs. Name the first one 'Zangband', and
call the second 'AMain', 'Winifred' or whatever takes your fancy. Once
your archive has been extracted, copy the contents of the 'bin' directory
to the 'AMain' disc. All the other bits go on the 'Zangband' disc.

Problems fitting everything onto that second disc? Don't worry. All the
files aren't essential. You can lose any of the sound files from
xtra/sound, or the help files which are cunningly hidden in /help.

Practically everyone has a hard drive in this day and age, but if you're
still running off floppies and having trouble persuading Zangband to
become floppy based, mail Mark and I'm sure he'd be more than happy to

```
help.

Playing Zangband
~~~~~~~~~~~~~~~~
If everything went well, and you successfully installed the game, then
all you need do is run the Zangband executable. Zangband automatically
loads in the previously saved character, but you can change this if
needs be using the -u command;

o Zangband -u<char>

So, if you wanted to load up Sharon the Half-Troll Warrior, you'd type
the command Zangband -uSharon. Simple, huh? Typing Zangband -? will give
you a list of the options available.
```

1.5 Settings

```
Settings
~~~~~~~~
Angband and it's variants need to know how to set themselves up, what with
screenmode, graphics, sound, menus, windows - it's all user configurable.
The configuration for this is read from the settings.prf file, which can
be found in the /user directory. Below is a brief description of the
various settings available and how they can help to make your gaming
experience that little bit easier. All of the settings are optional, and
have their own default settings.

o ANGBAND.gfx
  Do you want graphics or just plain ASCII?  Rogue old timers and users
  that find the graphics version a tad slow on 68000 machines may well
  benefit from setting this to Y.  Set to N for ASCII, Y for 8x8 bitmap,
  E for 16x16 bitmap. If E is selected and a suitably sized screen is
  not used, then the graphics will be autoscaled according to the font
  size for the MAIN window set in the settings.prf file. Should you get
  sick of the graphics and long for a return to ASCIIdom, they can be
  turned off in game from the pull down menus.

o ANGBAND.sound
  Do you want to include sound in the game?  Floppy users will have no
  choice but to leave sounds out.  Individual sounds can be removed, see
  the section of the documentation on sound for that.  Set to Y or N.

o ANGBAND.version
  The version of the game that you're playing. A setting used by the
  AngMan program for determining the type of variant being played.
  Optional.

o SCREEN.blankmouse
  Blanks the mouse pointer as and when a key is pressed.  Saves mouse
  pointer getting in the way whilst you're playing the game. Set to Y or N.

o SCREEN.menus
  Use the pull down menus or not?  Set to Y or N.  If you choose to use
  the pull down menus, they can be configured via the config file
  xtra/cfg/menu.cfg. Instructions for the format of this file can be
```

found at the top of it.

o SCREEN.aga
  Quick graphics – can speed things up on 68020+/AGA machines. Set to Y/N.
  Used to be called SCREEN.quick, and this can still be used.

o SCREEN.name
  If you've made a custom screen for Angband to display on, then you
  provide the name of the screen after this setting.

o SCREEN.mode
  Choose the screenmode that the Angband public screen will open under.
  Set to the mode value or ? to have a screenmode requester pop up.

o SCREEN.width
  Width of the screen in pixels. Minimum default of 640. You don't need
  normally need to set this or SCREEN.height.

o SCREEN.height
  Height of the screen in pixels. Minimum default of 200.

o SCREEN.depth
  Depth of the screen in bitplanes.

o SCREEN.overscan
  Screens overscan settings. Set to 0/1 for text size, 2 for graphics size,
  3 for extreme size and 4 for maximum size.

Window Options
~~~~~~~~~~~~~~
Angband and it's variants has the choice of using up to eight different
windows to display all of it's information. The use of MAIN is an
absolute necessity, but the use of CHOICE, RECALL, MIRROR, WIN5TH, WIN6TH,
WIN7TH and WIN8TH are all optional. They come into their own when used
with high screen resolutions, as they can be used to constantly display
information which you would otherwise have to access via options – keep
a list of your inventory in one, a list of your equipment in another, a
list of all past messages in a third – the choice is yours. What the
window actually displays is set from within Angband itself via the options
menu (= command). Their settings and positions can also be saved from the
pull down menus in Angband too.

MAIN Window
~~~~~~~~~~~
o MAIN.use
  Should the main window be displayed? Should be set to Y at all times,
  otherwise you've no chance of seeing a single thing!

o MAIN.title
  The main windows title, to be displayed in the title bar if on a custom
  screen.

o MAIN.font
  Font to use with the main window. Defaults to topaz/8.

o MAIN.xpos
  X-axis offset for the main window from the top left of the screen. Set

        with number of pixels.

    o MAIN.ypos
      Y-axis offset for the main window from the top left of the screen. Set
      with number of pixels.

    o MAIN.cols
      Number of text columns that the main window has. Maximum value of 80.

    o MAIN.rows
      Number of text rows that the main window has. Default value of 25.

    o MAIN.background
      Path to an file that is used as a background pattern for the main
      window. The 68K versions also support pattern loading via datatypes. PPC
      datatype support will come soon.

Other Windows
~~~~~~~~~~~~~
In the example below, I've used the window CHOICE, but this can be
substituted for any of the remaining windows as they're all defined in
exactly the same way.

    o CHOICE.use
      Same as for MAIN. Should the window be used? Set to Y or N.

    o CHOICE.show
      Should the window be displayed? Set to Y or N.

    o CHOICE.title
      Same as for MAIN. The title of the window to be displayed.

    o CHOICE.font
      Same as for MAIN. The choice of font that will be used in the window.

    o CHOICE.scroll
      Should the window have a scroll bar? Set to Y or N.

    o CHOICE.xpos
      Same as for MAIN. X-axis offset from the topleft of the screen.

    o CHOICE.ypos
      Same as for MAIN. Y-axis offset from the topleft of the screen.

    o CHOICE.cols
      Same as for MAIN. Number of columns of text to display.

    o CHOICE.rows
      Same as for MAIN. Number of rows of text to display.

    o CHOICE.background
      Same as for MAIN. Choose a background pattern.

Borg Options
~~~~~~~~~~~~
A Borg is a piece of code that will play Angband or it's variants all on
it's own. If your current version of Angband has been compiled with a Borg

included, then you will be able to use the following options;

o BORG.turns
  How frequently the display is updated, in turns. Can be set to anything
  up to every 500 turns.

o BORG.keycheck
  How the user can interrupt the Borgs game. Can be set to delayed,
  normal or joybutton.

1.6 Inscriptions

```
Inscriptions
~~~~~~~~~~~~
```
Inscriptions are a pretty useful part of Angband and it's variants, so
long as you know how to make the most of them. Once you've been playing
the game for a bit, you'll be used to the standard inscriptions that
objects have – telling you when they're cursed or uncursed, your pseudo-id
kicking in to announce that an object is terrible, worthless, average,
good, excellent or special. Shopkeepers will inscribe items with any
special offers they might have going. You might even get imaginative
yourself and decide that Brian the Broadsword should have his name
enblazoned across his blade. Well, written in braces next to his
inventory slot!

```
Protecting Items
~~~~~~~~~~~~~~~~
```
Inscriptions can help you protect your items from any inadvertant
keypresses your impatient fingers may be capable of. Once you become adept
at the controls of Angband, and could guide a character with your eyes
closed, or in your sleep, you'll soon discover that it's all too easy to
destroy, drop, activate, fire, etc that one item that you just can't do
without. How do you get around this? Use the following inscription;

o !X where X is the command to protect the item from.

So, if you inscribe an item with !d, you'll be asked if you really do want
to drop that particular object. If one check isn't enough, you can add
multiple inscriptions of the same type – inscribing !d!d!d will ask you
about dropping the object three times over. You may think it overkill, but
you don't want to go dropping plenty of things in the lower depths of the
dungeons of Angband.

You can combine the !X commands – if you inscribe !d!k, then you'll be
asked again if you try to either drop or destroy the item. If you need
FULL protection, you can use !*, which will ask for verification no matter
which command is used upon the item in question.

Roguelike Keyset Users – you'll have to inscribe the original commands, not
the roguelike ones, unfortunately.

```
Numbering Items
~~~~~~~~~~~~~~~
```
Inscriptions of the following form are incredibly useful when combined

with macros. They assign a number to a particular item, thus;

o @n where n is a number between 0 and 9.

What use is this? Well, say you have a particular wand or staff that you
use all the time. Instead of having to check your inventory to see which
slot it's in when you want to use it, you could inscribe it with @1 which
would let you select it by using the number 1 in with your command. You
can also restrict what commands use the numbered item;

o @Xn where n is a number between 0 and 9, and X is a command.

This will cause the item to only be referred to as this particular number
by the command X. A fine example of this is the one macro that's
automatically defined in all Angband variants – if you inscribe your main
weapon as @0, and your shovel, pick or secondary weapon as @0 also, the
command X will swap between these two items for you.

1.7 Macros

Macros & Keymaps
~~~~~~~~~~~~~~~~~
Macros are mappings from a single logical keypress to a sequence of
keypresses, allowing you to use special keys on the keyboard such as
function keys or the keypad, to automate repetitive multi–keypress
commands that you use alot.

The macro system has changed between Angband 2.8.2 and 2.8.3, and some
older variants are still based on the 2.8.2 code. I shall concentrate on
the 2.8.3 code here, and direct users of the older system to the file
help/command.txt, which should contain details on the old macro system.

Macro and keymap definitons are kept in what are called user pref files,
which can be found in the user/ directory, funnily enough. Specific .prf
files are loaded in with each character – if you run a human priest called
Kenneth, then the files human.prf, priest.prf and Kenneth.prf would all be
loaded. You can thus have similar macros and keymaps together – spell
macros could be in priest.prf, and macros you wanted humans to have in
general could go in human.prf, and any specific to just this character –
say to activate a particularly good item they have – would go in
Kenneth.prf.

So, how do you define macros and keymaps? Both of them need an action;

o A:<action> – this is the definition of the action the keymap/macro
               will expand to.

To define a macro to perform the above action you would follow the action
definiton line with;

o P:<str> – str is the sequence of keypresses that would perform
            the action.

If, however, you wanted to define the above action as a keymap, then you'd
follow it's definition with the line;

    o C:<mode>:<keystr> - mode is the keymap mode where the keymap will take
                          effect (0 for original and 1 for roguelike) and
                          keystr is a string that translates to one key. Note
                          that Ctrl-<char> is considered one key, but function
                          keys aren't.

So, a macro is defined by;

o A:<action>
  P:<str>

and a keymap by;

o A:<action>
  C:<mode>:<keystr>

Whenever macros need to contain special characters, they are represented
in a specific way, thus;

o \xHH - character number HH in hexadecimal
o \NNN - character number NNN in octal
o \e - ESC
o \n - newline
o \r - RETURN
o \s - space
o \ - backslash
o \^ - caret
o ^X - code for any control key "ctrl-X"

So, what's the difference between the two? Why choose a macro definition
over a keymap definition, or vice versa? There are three main differences;

o Keymaps don't get executed in stores, macros do. Imagine having 'u' set
  up as a macro, then trying to sell item 'u' in your inventory - as you
  hit u, up sprouts your macro string! You can't sell item u!

o Keymaps are defined as one key, but macros can be triggered by an
  arbitary number of keys. As function keys are considered more than one
  keypress, anything you bind to a function key has to be a macro.

o Keymaps can be different depending on the mode - original or roguelike.
  Macros use original commands.

Examples; well, ones that I use..

Activate light source - this is a macro which uses F1 for when I have the
Phial of Galadriel. It can't be defined as a keymap as the code for the
F1 key isn't a single key. Not that it stops anyone else using another
key, or course!

o A:\e\eAf\e
  P:\x9B0~

Kill top item on the floor - I played the 'bands for SO long without
knowing how useful this little one could be. If you've got a pile of stuff
on the floor, and you know the top three items are useless, just stand on

it and hammer the key three times, and they're destroyed. Very useful for
going through good sized treasure hoardes.

Macro;

o A:\e\e^d-yy\r
  P:`

Keymap;

o A:\e\e^d-yy\r
  C:0:`

I use a keymap in the above example, as a) it's simpler, and b) you don't
have to rely on the ` key in stores whatsoever. I map it to the ` key as I
use the roguelike command set, and so I can hit it with my left hand whilst
controlling the character with my right. If you use the original command
set, and you use the keypad, you might find it easier to maybe set up F10
as the macro. Why all the \e in the commands? It's to flush the output
buffer of any text it might display, and to get rid of any -more- messages.
Some people use \s too, as it performs the same task, so long as you have
quick messages activated in your options.

Example using numbered inscriptions;

Say you have a Rod of Trap Detection inscribed with @z1, and a Rod of
Detect Doors and Stairs inscribed with @z2. You could combine the
activation of the two by pressing F1 by setting up the macro;

o A:\e\ea1\ea2
  P:\x9B0~

\e\e will clear the output buffer of any existing messages that have yet
to be displayed, a1 will activate rod 1 (trap detection, irrespective of
it's location in the inventory), \e will clear the result of the
activation of rod 1, a2 will activate rod 2 (doors and stairs). Easy, huh?

1.8 Variants

Variants
~~~~~~~~
Sure, Angband is a great game, but there are people out there who want
more. People who want to personalise the game, make their own alterations,
or take it in a completely new direction. If they're lucky enough to be
familiar with programming C, then all they need do is alter the source
code and a new variant is born.

What follows is an exhaustive list of the variants available. Unfortunately
(fortunately in some cases) they're not all available for the Amiga. Those
that are can be found in bold type. Check out the links section of the
documentation if you'd like to know more about any of the variants below.

o Angband/64 - lots more traps, bigger dungeons, reformed spell system.

o AngbandTk - Angband with a swanky graphical system written in TCL 8.0.

o Bangband – Now defunct variant letting the player be dragons and monks.

o Cathband – Another oldie, aiming to balance all classes.

o Cthangband – Zangband variant, replacing Zangband's Zelazny world with that of H.P. Lovecraft's Cthulhu mythos.

o Drangband – Ancient variant allowing various dragons as players.

o Fangband – First variant to include targetting code for bows!

o Goingband – Lots of patches written by various people.

o GSNband – Zangband variant, removing all Zelazny material in order to get back to the Tolkien feel of the game.

o GW-Angband – New magic system, balanced classes.

o Ingband – Ironman Angband – dive 'til you drop!

o Kamband – Kangband variant – religions, new races, alchemy, new spells.

o Kangband – New classes, bigger town, wilderness area around the town.

o KangbandTk – Kangband with the TCL 8.0 graphical interface.

o Langband – Work in progress – very configurable.

o Mangband – Multiplayer Angband, played via a TCP connection.

o Oangband – Opinion Angband – changes right across the board.

o Pangband – Plot based Angband, uses RPL scripts to describe the plot.

o PernAngband – Zangband variant based on the novels on Anne McCaffrey.

o Pziband – Formerly Psiband; Angband with psionic powers.

o QuestAngband – Variant based around a quest system.

o Rangband – Realistic Angband. Reality steps into Angband.

o SBFband – Steve Fisher's own version of Zangband.

o Sangband – Skills based Angband. Develop various skills to advance.

o Yin-Yangband – Various changes to classes, races etc.

o Zangband – Most played game, even moreso that Ang. Changes galore!

o ZangbandTk – Zangband with the TCL 8.0 graphical interface.

1.9 Links

```
Links
~~~~~
Here are a list of useful Angband related websites, along with a blatant
plug for my own site!

o Newsgroup
  rec.games.roguelike.angband

o Official Angband Page
  http://www.phial.com/angband

o Bablos' Angband Variants Page
  http://wkweb5.cableinet.co.uk/bablos/angband

o Development FTP Site
  ftp://export.andrew.cmu.edu/angband

o Roguelike News
  http://www.skoardy.demon.co.uk/rlnews

o Statistics, Highscores and Analysis for Roguelike Derivatives (SHARD)
  http://www.frosch.org/shard.html


Variant Homepages
~~~~~~~~~~~~~~~~~
o Angband/64
  http://www.sx4all.nl/~thunder7/angband

o AngbandTk
  http://www.mypage.direct.ca/d/dbaker/angbandtk.html

o Cthangband
  http://cthangband.8m.com

o GSNband
  http://ifg.amu.edu.pl/~naskrent

o GW-Angband
  http://www.kellnet.com/wooledge/angband.html

o Ingband
  http://www.sucs.swan.ac.uk/~element/ingband

o Mangband
  http://www.umr.edu/~keldon/mangband

o Oangband
  http://home.sprintmail.com/~leon2m/Oangband_Development_Site.htm

o Pangband
  http://www.umr.edu/~keldon/pangband

o PernAngband
  http://perso.infonie.fr/dark.godZangband
```

```
o Quest Angband
  http://pet.softjoys.ru/~qa

o Sangband
  http://www.fragment.com/~jl8e/sangband

o SBFband
  http://www.cs.unr.edu/~fisher_s/sbfband.html

o Zangband
  http://thangorodrim.angband.org
```

1.10 History

```
Credits
~~~~~~~

o Mark Howson <markh@angband.org>
  Mark maintains the best ports of Angband and it's variants that
  are currently available. Mark took on development from Lars'
  2.7.9v6 Angband code, and things have just grown exponentially
  since then. In amongst the Amiga ports, he also manages to be the
  official maintainer of Kangband, believe it or not!

o Bablos <bablos@cableinet.co.uk>
  Me. I maintain a number of variants, thanks to Mark's excellent
  help. I wrote this documentation (well, the bits that I didn't rip
  off Mark's initial efforts!), I upload the variant archives to
  Aminet, I beta test both 0x0 and PPC versions, and I run the
  Angband Variants Website - I'd best not plug it again in these
  docs, as Mark will want to hit me with a large stick...

o Ben Harrison <benh@phial.com>
  Ben is the current maintainer of Angband. See the official website
  at http://www.phial.com/angband for more information on the history
  and current status of Angband.

o Robert Ruehlmann <rr9@angband.org>
  Robert is the current maintainer of Zangband. He took over from Topi
  Ylinen, who had to abandon the game due to real life taking over his
  spare time. Topi still has a great deal of input with regards to the
  direction of the game, though...

o Lars Haugseth <larshau@ifi.uio.no>
  Lars wrote the original Amiga Angband source code.

o Torben Kristiansen, James Larimer and Thomas H. Davies.
  The original Amiga Angband graphics were designed by these guys.
```

1.11 Contact

```
Contact Information
~~~~~~~~~~~~~~~~~~~
```

Got something to say? Like to buy Mark a pint or a Mars Bar for his
wonderful work? Head full of suggestions, questions, comments,
criticism, improvements or insults? If so, Mark wants to hear it.

Start up your favourite email client and drop him a line at;

o markh@angband.org

Mark wants people to understand that he didn't actually write any
of the ports available, he just converts them for our use. Any bugs
that are obviously specific to the code and not the Amiga version
should be directed towards the original author via email or the
newsgroup.

Should you have the need to contact my good self, be it suggestions
for additions to the documentation, questions regarding the Aminet
uploads, or adoration of my web page, you can find me at;

o bablos@cableinet.co.uk

1.12 The Borg

```
The Borg
~~~~~~~~
```
The Borg is an automatic Angband player, ie. the computer will play
Angband and it's variants as if it was a human player, making decisions
on its own.

o The borg will not be getting support - Mark says "I'll fix the Amiga
  code bugs, and I'll look into fixing main source code ones (or report
  them to Ben), but I don't want to get into debugging the thing."

o The borg needs at least 2Mb of memory, and at least a 68020 to be
  usable.

What follows is a list of the basic commands for the borg. Should you
want to know more, I suggest you aim your queries at the borg source
code.

Use Ctrl-Z to activate the borg, reply yes to the warning messages, and
then press 'z' to start the borg running. It will stop when you press a
key, so you can play the game yourself for a while and then let the borg
takeover.

o Note - You can't have graphics & the borg at the same time - you must
  turn graphics off first. If you can, open up a couple of extra windows
  for the borg messages/status.

The rest of the borg commands are documented internally - type Ctrl-Z ?
for a simple summary. But I will mention here the 'f' command, which can

be used to save the borg whenever it changes level or depth. Use 'fs' or
'fd'.

There are a great many more commands than this – refer to 'borg9.c' for
a more thorough explanation. If you can't read C, get in touch with Mark.

Oh, and the borg does seem to crash occasionally. I've left it running
overnight without problems, but it may hang once or twice. You have been
warned – don't leave anything really important unsaved in the background.
I'm sorry about this, but I think this is a borg bug, rather than an
Amiga one. And whatever it is, it'll be VERY hard to find. Don't let that
put you off – it seems to be fairly stable. More so than some Amiga
programs I could mention!