

Kangband

Bablos <angband@blueyonder.co.uk>

COLLABORATORS

	<i>TITLE :</i> Kangband		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Bablos <angband@blueyonder.co.uk>	August 10, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Kangband	1
1.1	Kangband Documentation	1
1.2	Introduction...	1
1.3	Requirements...	2
1.4	Installing Kangband	3
1.5	Settings	4
1.6	Inscriptions	7
1.7	Macros	8
1.8	Variants	10
1.9	Links	12
1.10	History	14
1.11	Contact	14

Chapter 1

Kangband

1.1 Kangband Documentation

Kangband 2.9.2r2

~~~~~

Introduction

Requirements

Installation

Settings

Inscriptions

Macros

Variants

Links

Credits

Author

Written by Bablos, 22.04.2001

### 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.

Kangband

~~~~~

Kangband was originally developed by Ken Wagle, but was taken over by Mark Howson when Ken had an attack of Real Life(tm). Mark abandoned Kangband during Python integration development, and John I'anson Holton came along to pick up the pieces. Kangband was the first Angband variant to expand the town and associated terrain - the town expanded to include castles, gambling houses, and even it's own surrounding wilderness. The wilderness idea soon spread to the dungeons themselves, being populated with all manner of flora and fauna. Kangband was responsible for spawning the first variant of a variant - Kamband.

For full details of all the changes made to Kangband throughout it's history, please consult the changes.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 Kangband.

- 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. There is no 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 and I have put together for those users who aren't living in the stone age...

Machines that Kangband have been tested on;

- o A1200, 68060/50, 603e/200, 32Mb FAST, AGA, OS 3.1 (Bablos)
- o A1200, 68060/50, 603e/200, 32Mb FAST, AGA, OS 3.9 (Bablos)

## 1.4 Installing Kangband

Installing Kangband

~~~~~

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 Kangband up and running is to run the program Kangband.

- 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 Kangband. Anyway, the commands required are;

```
assign Kangband: DH1:Games/Kangband
path add Kangband:
```

Obviously, replace the DH1:Games/Kangband with the path to the Kangband directory on your system. Make sure that this new Kangband: 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 Kangband 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 'Kangband', and call the second 'AMain', 'Winifred' or whatever takes your fancy. Once your archive has been extracted, copy the executable to the 'AMain' disc. All the other bits go on the 'Kangband' 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 Kangband to become floppy based, mail Mark and I'm sure he'd be more than happy to help.

Playing Kangband

~~~~~

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

o Kangband -u<char>

So, if you wanted to load up Sharon the Half-Troll Warrior, you'd type the command Kangband -uSharon. Simple, huh? Typing Kangband -? 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.

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.

- o SCREEN.quick
Quick graphics - can speed things up on 68020+/AGA machines. Set to Y/N.
- o SCREEN.rtg
Use RTG system for screen display? Set to Y or N.
- 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 public screen in pixels. Minimum default of 640.
- o SCREEN.height
Height of the public screen in pixels. Minimum default of 200.
- o SCREEN.depth
Depth of the public screen in bitplanes.
- o SCREEN.overscan
Screens overscan settings. Set to 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. Must be set to at least 80.
- o MAIN.rows
Number of text rows that the main window has. Must be set to at least 80.
- o MAIN.background
Path to a file that is used as a background pattern for the main window. This option is only active in current versions of Angband and it's variants.

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.
-

## 1.6 Incriptions

### 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 inadvertent 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>:<keyst> - mode is the keymap mode where the keymap will take effect (0 for original and 1 for roguelike) and keyst 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>:<keyst>

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 arbitrary 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 Amberband - Angband with a heavy Amber influence.
- o Angband/64 - lots more traps, bigger dungeons, reformed spell system.
- o AngbandTk - Angband with a swanky graphical system written in TCL.
- o Bangband - Now defunct variant letting the player be dragons and monks.
- o CathAngband - aiming to balance all classes in Angband.
- o Cthangband - Zangband variant, replacing Zangband's Zelazny world with that of H.P. Lovecraft's Cthulhu mythos.
- o Discband - Terry Pratchett's Discworld, in Angband form.

- o Drangband - Ancient variant allowing various dragons as players.
 - o DvEband - Angband with Oangband influences.
 - o Easyband - For those who have trouble getting past 50'.
 - o EyAngband - Eytan Zweig's own Angband adaptation.
 - o Fangband - First variant to include targetting code for bows!
 - o FFVII Angband - Variant based on Squaresoft's excellent RPG.
 - 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 GSN2band - Vanilla based GSN minus the Zangband elements.
 - o Gumband - Joseph Dixon's own adaptation of Zangband 2.1.1c.
 - 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 graphical interface.
 - o Langband - Stig E. Sandø's Common Lisp re-write of Vanilla.
 - o Mangband - Multiplayer Angband, played via a TCP connection.
 - o MJBand - Angband variant with some interesting changes.
 - o NewAngband - PernAngband with updated class system.
 - o Oangband - Opinion Angband - changes right across the board.
 - o OangbandTk - Oangband with the TCL graphical interface.
 - o Pangband - Plot based Angband, uses RPL scripts to describe the plot.
 - o PernAngband - Zangband variant based on the novels on Anne McCaffrey.
 - o PernMangband - Fusion of PernAngband and Mangband.
 - o Psiband - Angband with psionic powers.
 - o Pziband - Zangband with psionic powers.
 - o QuestAngband - Variant based around a quest system.
 - o Rangband - Realistic Angband. Reality steps into Angband.
-

- o Sangband - Skills based Angband. Develop various skills to advance.
- o SBFband - Steve Fisher's own version of Zangband.
- o Sillyband - Angband with ridiculous alterations.
- o Tangband - TeamAngband, taking the traditional RPG approach to adventure.
- o Terminal Angband - Beta of Angband using multiple term windows.
- o TKKband - South Park derived variant.
- o Yakband - bizarre variant, where you can play a Yak, of all things!
- o Yin-Yangband - Various changes to classes, races etc.
- o Zangband - Most played game, even moreso than Ang. Changes galore!
- o ZangbandTk - Zangband with the TCL 8.0 graphical interface.
- o Zceband - Eric Bock's own version of Zangband.

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](mailto:rec.games.roguelike.angband)
- o Official Angband Page  
<http://www.phial.com/angband>
- o Bablos' Angband Variants Page  
<http://www.angband.pwp.blueyonder.co.uk>
- o Development FTP Site  
<ftp://export.andrew.cmu.edu/angband>
- o Roguelike News  
<http://www.skoardy.demon.co.uk/rlnews>

### Variant Homepages

~~~~~

- o Amberband
http://members.xoom.com/Lord_Emrys/amber.htm
- o Angband/64
<http://www.sx4all.nl/~thunder7/angband>
- o AngbandTk

- <http://www.mypage.direct.ca/d/dbaker/angbandtk.html>
- o CathAngband
<http://dbox.victoly.com/~neko/cath>
 - o Cthangband
<http://www.visi.com/~thornley/david/cthangband>
 - o Drangband
<http://hercules.itctel.com/~apwhite/drang.html>
 - o EyAngband
<http://eyangband.sourceforge.net>
 - o GSNband
<http://ifg.amu.edu.pl/~naskrent>
 - o GW-Angband
<http://www.kellnet.com/woledge/angband.html>
 - o Kangband
<http://kangband.sourceforge.net>
 - o Langband
<http://langband.sourceforge.net>
 - o Mangband
<http://mangband.org>
 - o MJBand
<http://www.litening.dircon.co.uk/mjband.html>
 - o NewAngband
<http://members.nbci.com/variav>
 - o Oangband
<http://www.consume.org/Oangband>
 - o Pangband
<http://www.umr.edu/~keldon/pangband>
 - o PernAngband
<http://www.pernangband.org>
 - o PernMangband
<http://pernmangband.pernangband.org>
 - o Sangband
<http://www.fragment.com/~jl8e/sangband>
 - o SBFband
http://www.cs.unr.edu/~fisher_s/sbfband.html
 - o Sillyband
<http://www1.50megs.com/tarael>
 - o Zangband
-

<http://www.zangband.org>

1.10 History

Credits

~~~~~

- o Bablos <angband@blueyonder.co.uk>  
Me. I maintain every variant for the Amiga these days, apart from Richard Smith's Mangband and PernMangband. I wrote the Amiga specific documentation, I beta test all variants, I upload any Aminet archives that are ready, I run the Angband Variants Website, and I run lots of borgs on my machine!
- o Mark Howson <markh@angband.org>  
Mark took over the Amiga specific Angband code from Lars Haugeth's 2.7.9v6b code, and improved it alot up until 2.8.2. Since then Mark's machine has died, and real life prevents him from developing and compiling further, so I took on that part of things. Mark has moved on from the Angband community now, but he still responds to any code related questions I care to throw at him.
- o Rob Reuhlmann <rr9@angband.org>  
Rob took over the Angband maintainership from Ben Harrison in April 2000, and released 2.9.0 as his mark on the world of Angband. He also heads the Zangband development team, taking over it's development from Topi Ylinen.
- o <variant author> <variant authors email address>  
<variant author> is responsible for creating <variant>
- o Andrew Tait <antait@blueyonder.co.uk>  
Totty is the main beta tester besides myself, and might end up being the only other Amiga user to actually play some of the variants I knock out! He runs lots of borgs on his machine, too...

## 1.11 Contact

Contact Information

~~~~~

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

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

- o Bablos
angband@blueyonder.co.uk

- o Mark
markh@angband.org

Please remember however, that neither of us actually write any of the variants that are currently available, we just convert them for use on the Amiga. 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.
