

HyperHelp

Sean Russell

COLLABORATORS

	<i>TITLE :</i> HyperHelp		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Sean Russell	November 24, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	HyperHelp	1
1.1	HyperHelp	1
1.2	Notes about the versions	1
1.3	Welcome to HyperHelp	2
1.4	Getting it all started	4
1.5	Configuring HyperHelp to your system	5
1.6	The Preferences program	8
1.7	Index	9
1.8	The Stream interface, Watchdog	10
1.9	priority	11
1.10	buffer	11
1.11	Stream Buffer	11
1.12	HotKey	12
1.13	Edit...	12
1.14	Force Key	12
1.15	Edit Force	12
1.16	Editor	13
1.17	Port	13
1.18	Command	13
1.19	Save	13
1.20	Quit	14
1.21	Build Index	14
1.22	The index file	14
1.23	The commodities program	15
1.24	Rights and reservations	16
1.25	The CLI index creator	18
1.26	The CLI preferences program	18
1.27	Contacting the programmers	19
1.28	Bugs	20
1.29	General Q&A, and paradigms	21
1.30	About Magic User Interface	22
1.31	Credits	22
1.32	You did WHAT?!?!?	22

Chapter 1

HyperHelp

1.1 HyperHelp

HyperHelp

Release 1.01

1994 Apocalypse Productions

Helmut Lutzenberger

Sean Russell

Index

Version notes.

Introduction Sytem requirements and overview

Installation

Configuration

HyperHelp The preferences program

HyperHelp_cx The commodities program

MakeIndex The CLI index generator

MakePrefs The CLI preferences generator

Copying What you can and can't do with this package

Authors How to contact the programmers

Bugs ...or at least the ones we know of

Notes on implementation

MUI The reason for the wonderful GUI

Thanks to...

Commodore

1.2 Notes about the versions

The versions for the various programs in this archive are different, since each program is developed separately. For this reason, this

archive is supplied a completely separate release number, and it is this number which is considered the version of the HyperHelp package. For instance, at the time of the initial writing this document, **HyperHelp** (the preferences program) has a version of 0.5; **HyperHelp_cx** (the commodities program) has a version of 0.83; however, when we say, for example, that this is version 1.00 of HyperHelp, we are referring to the release number of the archive. The release number will always be the same as the version number of this file.

The version of HyperHelp.guide (this document) represents the release number of the archive.

See the CRC History for CRC checks.

You can compare the dates in the CRC history file with the dates in the archive ("lha l hyperhelp.lha"), but remember that if some moron really wants to infect this archive with a virus, your only recourse is a good virus checker, or calling (or E-Mailing) us for CRC validations.

If you feel any reason to doubt these values, contact either of the authors for validation. In this case, please be prepared to give the version of the individual program, not the version of the HyperHelp archive (of this document).

1.3 Welcome to HyperHelp

The HyperHelp package is a collection of files designed to maintain and provide access to an AmigaGuide(TM) (HYPERtext) based hotkey HELP system.

There are already a couple of commercial hotkey-based help systems available, (one of them just recently went semi-PD). Here is a list of things HyperHelp does which make it cooler than the others:

IT'S FREE! Or, it could be. It could also cost you money. That depends on your conscience. The point is, you didn't pay any money to get this.

Fully AmigaGuide based. All help texts are AmigaGuide files, which can be easily added to your help base with a drag-and-drop (or ASL file requester) approach. This means no silly conversion programs or special data formats.

And you will benefit from all improvements in the AmigaGuide library.

A cool MUI preferences program. (And plenty of CLI support for you Shell people).

Asynchronous help windows. You can have as many help windows open at one time as your RAM allows.

Intelligent hotkey parsing. HyperHelp_cx doesn't just perform a preset action when you press a key. HyperHelp will try to draw a keyword from the environment you are working in, whether it be a shell, a string gadget, or a text editor.

Levenshtein algorithm. HyperHelp_cx uses a sophisticated pattern matching algorithm. This means that HyperHelp_cx will select not only exact matches, but will also allow you to choose from similar matches.

HyperHelp_cx supports wildcards. You can pattern-match with '?' (any character) and '*' (any number of any characters).

HyperHelp can be configured to work with almost any text editor which has an ARexx port.

Goes easy on your system resources (not too much memory, almost no much CPU time).

Multicultural. (What does that mean?!?). Hey, you get support in the US and Europe.

Easy to install. All we supply are the executables; they don't have to be in any particular directory... there are no assigns you have to make... just point and click! (Ok, ok... you do have to configure it first, but nothing is perfect)

It's fast. On a 68EC030 with no FPU, AT/IDE harddrive (+10k HyperHelp buffer) _Cx searches a 42k file (around 2963 nodes) in ~2 seconds.

People will respect and admire you. Like many brands of beer, some kinds of cigarettes, and all brands of snack foods, HyperHelp will make you more attractive to the opposite sex and the center of attention at parties. It's true, really. I saw it on TV.

The archive consists of four primary files:

HyperHelp - The preferences and maintainance program.

HyperHelp_cx - The commodity program.

MakeIndex - A program to generate **indexes** from the CLI.

MakePrefs - A program to generate a

preferences file from the CLI.

In addition, there are the actual guides, **index** files and preferences files which are either supplied by the user (you), or are generated by either the preferences program **HyperHelp** or the CLI support programs. Reading the section on **installation** should provide all the necessary information required to get the HyperHelp package running on your system.

You need at least OS2.0 (v37+) to run the HyperHelp system. In addition, the GUI preferences program requires muimaster.library v4 or better. For more information on this library, see the section on **MUI**. We are not allowed to distribute the MUI library (on request of the author of MUI); however, the MUI package is freely available. Obviously, you will also need amigaguide.library; any version after v34 will do. This is also a freely distributable library. We did not include it with this package since almost everybody already has the library, and it would only increase the size of the HyperHelp archive.

HyperHelp_cx waits in the background for you to press one of the hotkeys (default: right-shift-help or right-alt-help). When one of these events occur, HyperHelp_cx will perform some action based on the hotkey you selected. If you press the standard hotkey, HyperHelp_cx will search through an **index** file (which you create with the help of a supporting file) for the keyword and opens an AmigaGuide file with information over that keyword. If you choose the Force hotkey, you force HyperHelp_cx to open an index of all files it has access to.

1.4 Getting it all started

- Overview

These are the things you must do to get the HyperHelp system running (clarified later in Configuration):

- 1 - Set the **index** path and name
- 2 - Set the editor port and command
- 3 - Build an index file

- Installer

The installer file included with the package should make things almost effortless. For this to work, however, you must have the Commodore

Installer program (freely available). If you choose to install by hand, this is what the script does:

Copy **HyperHelp** (and the icon) into the prefs/drawer of your system disk.

Copy **HyperHelp_cx** (and the icon) into the WBStartup drawer of your boot (system) disk.

Copy **MakeIndex** and **MakePrefs** to C: (you don't need these if you have a registered version of MUI)

Copy this file into any appropriate drawer.

It (almost) couldn't be easier.

1.5 Configuring HyperHelp to your system

The configuration of the HyperHelp system requires a bit more work than the installation. Most of the configuration can be done over the MUI application **HyperHelp** (this is the preferences program). However, all options for **HyperHelp_cx** can also be set in the tooltypes of the HyperHelp_cx icon, or can be entered as command line arguments (in a script file, for example).

Note that HyperHelp_cx will first take preferences from a preferences file (if one exists), and then from the tooltypes (or CLI arguments), and then will use it's defaults.

HyperHelp_cx recognises the following tooltypes/arguments:

PORT/A/K - This is the name of the ARexx port your text editor makes available to you for process communication. You must find this port name in the manual of the text editor, or find it with a system-resource watcher (such as RSys). If you do not specify this argument, HyperHelp_cx will use the port EDWARD

(Maxon). Examples are: EDWARD, GOLDED.1, REXX_CYGNUS

COMMAND/K- This is the command to which your text editor will return the current word. This command (usually) tells the editor to return the current word under the cursor to the process making sending the command. If you do not specify this argument, HyperHelp_cx will assume GETWORD (for Edward and CygnusEd). Examples are: GETWORD, EXTRACT (GoldEd)

INDEX/K - This is the full path and name of the **index** file.

CX_PRIORITY/K/N - The priority of the commodity. Defaults

own. To set your own you must know the ARexx portname of the editor and the ARexx command to get a word. You must check the documentation of your editor to find these out.

5 - Press the Save button.

6 - Add at least one guide to the index file, either with the menu option "Add index..." or by dropping an AmigaGuide icon onto the "Build Index | (Drop Icon here)" box.

7 - Run the **HyperHelp_cx** program.

xx

You have an unregistered version of MUI:

1 - Run the **HyperHelp** program.

2 - Set the full path and name of the **index** file. You must do this every time you run this program. If this file does not exist it will be created.

3 - Add at least one guide to the index file, either with the menu option "Add index..." or by dropping an AmigaGuide icon onto the "Build Index | (Drop Icon here)" box.

4 - Set the tooltypes in the HyperHelp_cx icon to their appropriate values. Important are that you set the INDEX tooltype to be the full path and name that you specified in step 1, and that you set the PORT and COMMAND tooltypes. The PORT tooltype is the ARexx portname of your text editor, and COMMAND is ARexx command to request a word. You must check the documentation of your editor to find these out.

5 - Run the **HyperHelp_cx** program (from the workbench).

xx

You do not have MUI at all

1 - Run the MakeIndex program. You must give the full path and name of the **index** file as an argument.

Add at least one guide to the **index** file.

2 - Set the tooltypes in the HyperHelp_cx icon to their appropriate values. Important are that you set the INDEX tooltype to be the full path and name that you specified in step 1, and that you set the PORT and COMMAND tooltypes. The PORT tooltype is the ARexx portname of your text editor, and COMMAND is ARexx command to request a word. You must check the documentation of your editor to find these out.

3 - Run the **HyperHelp_cx** program (from the workbench).

[illegible]

You are a CLI-o-Phile

1 - Run the MakeIndex program. You must give the full path and name of the **index** file as an argument.

Add at least one guide to the index file.

2 - Run the **HyperHelp_cx** program with at least the following arguments:

INDEX - The full path and name of the index you supplied in step number 1.

PORT - The ARexx portname of your text editor.

COMMAND - The ARexx command to request a word from your text editor.

1.6 The Preferences program

HyperHelp requires a stack of at least 6000 bytes. To be really safe, you should set the stack to 8000 if you are running HyperHelp from the shell or if you for some reason change the icon.

HyperHelp recognises one tooltype/argument:

HELPPFILE=<file>

This argument tells HyperHelp where to find this file (which you are reading right now). If you correctly set this option, you will be able to get use the Help key to get information on the gadgets in HyperHelp.

In the preferences program you may set all of the various arguments described in the configuration section of **installation**. In addition, the prefs program has a few additional perks to make life easier for you:

* A popup file requester for choosing the **index** file. Remember that this must be the full path and name of the index file. If the file in this string does not exist, HyperHelp will create a new index file when you choose to add guides.

* A popup list of editors. There are currently 3 preset editors in this list. Double-clicking to choose an editor from this list will also set the port and command arguments for HyperHelp. The preset editors are Edward (Maxon), CygnusEd (CygnusSoft), and GoldEd (Dietmar Eilert). If you do not use one of these editors, you must

set the port and command options by hand.

* An AppWindow box. Dropping Icons into this box (labeled "Build Index | (Drop icon here)") will add the file to the index.

* An ASL requester. You may choose the menu option "Add index..." which will call up an ASL file requester. You may multi-select the files you wish to add to the index with this requester. This is convenient for first-time generation of indexes. Note:

Adding many files or adding files to an already large index can take time. Be patient. A progress indicator is provided, but this indicator increments by file, not by time. If a large file is added to a large index, you may see no change in the progress indicator for some time.

* Full help-key support for information on each gadget. Press the "Help" key while the mouse is over a gadget and a window will appear with information about that gadget.

Note: The Save function does not work if you do not have a registered copy of **MUI**.

A full explanation of each gadget can be accessed here:

Index

Priority

Buffer

Stream Buffer

HotKey

Edit...

Force Key

Edit Force

Editor

Port

Command

Save button

Quit button

App window box

1.7 Index

This string gadget must contain the name (with full path) of the **index** file. For example, if you wish the index file to be located in your "Work:documents/" directory, and you wish the index file to be called "index.HYP" then you must enter

"work:documents/index.hyp" into this gadget. If the file does not exist, it will be created when you add guides to the index. If the file does exist, new guides added will be appended.

Only one index file can be used by **HyperHelp_cx** at a time.

The button to the right of this gadget will bring up an ASL file requester with which you may select the file to use as an index. When you first setup the HyperHelp system, choose a non-existent file as an index. Do not use an already existing file! If you do use an already existing file, **HyperHelp_cx** and **HyperHelp** will still work; however, the speed of the programs will decrease and the wear-and-tear on your drive will increase.

1.8 The Stream interface, Watchdog

Normally, if we know the correct port name and the correct command, we can request a word from a text editor. Most shells don't have an ARexx port, however, so we need another way of finding out what the user has been typing so that we can draw a keyword from this information. This is where WatchDog comes in.

WatchDog is a small child process of **HyperHelp_cx** that keeps track of all the characters you type in. When you press the hotkey, **HyperHelp_cx** checks to see if the active window belongs to your editor. If not, then it asks WatchDog to return what the user has been typing. **HyperHelp_cx** then looks at that information, which consist of every key the user has pressed (for the last X keystrokes). It then attempts to intelligently interperate any editing codes that appear in the list of keypresses, and generates a keyword from this. This method has the advantage that we can also extract keywords from string gadgets and other such one-shot sources.

It is important to remember that, although **HyperHelp_cx** can deal with some editing of a line, it is impossible to correctly interperate all editing codes. This is because only a few codes are standard editing commands (left, right, delete, and backspace). Most other codes that often occur (shift-left, alt-left, etc.) often vary from shell to shell. In addition, **HyperHelp_cx** is unable to deal with history buffers. What this means is that if you type in a word in a shell and then use the history buffer command <up-arrow>, **HyperHelp_cx** will not be able to extract that word from the history buffer.

HyperHelp_cx has to have some way of telling what defines a "word."

The following characters define the boundries of a word:

Spaces

Non-alpha-numeric characters

Returns

Quotes

Parenthesis

You may set the amount of keystrokes WatchDog keeps track of with the SBUFF option, or the **Stream Buffer** gadget in

HyperHelp. Remember that no matter how large the buffer is, HyperHelp will only look at the last word in the stream. Since it is unlikely that you will be typing in words longer than 64 letters long (including editing commands), the default of 64 is normally adequate. More than 128 for the stream buffer verges on absurd.

1.9 priority

You may set the Commodities priority of the **HyperHelp_cx** program with this gadget. Values for this gadget must range from -127 to 127. If you do not know what effect the CX_PRIORITY of a program has, you can leave this value at it's default.

1.10 buffer

The value in this gadget will effect the speed of the keyword searches, and represents the buffer for disk operations. In general, the higher the value, the faster the search. The amount of memory that **HyperHelp_cx** will use is equal to about 47k + BUFFER.

This number should be no less than 200, and numbers over 10000 are excessive (and will provide increasingly less speed improvement). 5000 is a good average number, and you can experiment. If you have a very slow hard drive, you might use a higher number (5000 +); a fast SCSI-II drive could have a lower number (300-2000).

1.11 Stream Buffer

The value here determines how big the buffer for the stream should be. This buffer is where the HyperHelp system will store your keystrokes so that it can tell what you've been typing into the shell, or a string gadget, or... whatever. It is unusual that this should ever be

more than 64 bytes, because it is unusual that you will have a keyword longer than 64 characters, including simple editing. 32 bytes is also a safe number. The only negative effect that can result from this value being too small is that your search musters will be truncated, and you'll get keywords like: "arianism" from "disestablishmentarianism". (Even this example has at most 24 bytes).

1.12 HotKey

With this you can set which hotkey will call **HyperHelp_cx**. The default is right-shift - help ("rawkey rshift help"). The word "rawkey" is not required, and you may use any legal Commodities hotkey.

1.13 Edit...

This cycle gadget gives you two options for controlling the editing window. This window is HyperHelp_cx's way of letting you edit your search word. HyperHelp_cx will always open this window when it can't get a word from either an editor or from the stream. You may either close this window, enter return, or enter a new keyword into the string gadget. If you choose the option Every Time with the cycle gadget, HyperHelp_cx will always open this window when you press the hotkey, even when it finds a keyword. In this case, the keyword will appear in the string gadget, where you can edit it before pressing return. If you choose No word HyperHelp_cx will only open the window when it can't get a keyword.

1.14 Force Key

With this string you define the hotkey combination which will force HyperHelp_cx to open a guide to all of the files the HyperHelp system has access to. The default here is "right-alt-help".

1.15 Edit Force

If you check this option, HyperHelp_cx will open the editing window whenever you press the **forcekey**. In this case the string gadget will be empty, and you may either enter a keyword to search for, press return with an empty string to call the root index, or close the window (which also calls the root index).

1.16 Editor

This string has no actual effect on the operation of **HyperHelp_cx**, and is here only for reference. This string may be set by choosing an editor from the Editor list (accessed by pressing the popup button to the right of the string gadget). To choose one of the preset editors, open the list and double click on one of the editor names. The Port and Command strings will be automatically updated.

1.17 Port

This is the name of the ARexx port of your favorite text editor. To find out what this portname is you must read the documentation of your text editor. The information will likely be under a section called "ARexx", "ARexx Interface", or something similar. You do not have to set this value if you use one of the pre-set editors.

Note The Port and Command arguments are usually case sensitive. If you type these in yourself, you must make sure that you type them in exactly as they appear in the documentation.

1.18 Command

HyperHelp_cx will attempt to request a word from your text editor. Normally, this word will be the word under the cursor in your editor. The string in this gadget is the command **HyperHelp_cx** will send to your editor to request the word. You must check the documentation to find out what this value is. The command for Edward and CygnusED is "GETWORD" and for GoldED "EXTRACT", for example. You do not have to set this value if you use one of the pre-set editors.

Note

The Port and Command arguments are usually case sensitive. If you type these in yourself, you must make sure that you type them in exactly as they appear in the documentation.

1.19 Save

Press this button to save your configuration. This button only has effect if you have a registered version of **MUI**.

1.20 Quit

Pressing this button will end the **HyperHelp** program.

1.21 Build Index

Dropping the icons of AmigaGuide documents on this box will add the guide to the **index** file.

1.22 The index file

The index file is very important to the HyperHelp system. It is produced by either the MakeIndex (CLI) or the **HyperHelp** (MUI) program. Both of these programs search through the supplied (by you) AmigaGuide documents for nodes (buttons), which they then add to the index file.

The **HyperHelp_cx** program accesses this file whenever you press the hotkey and uses it to find out where each node is located.

HyperHelp_cx either tries to find a match for the supplied keyword (if any), or simply shows you all files (not all nodes) you have stored in the index.

The index file is a simple text file, and has the following format:

```
@ $ amigaguide_file_with_full_path
@ @node_name_1
@ @node_name_2
...
@ @node_name_last
@ $ amigaguide_file2_with_full_path
@ @node_...
```

...and so on. Blank lines and random text in this file are accepted; however, this will obviously increase the amount of time to search for keywords in the text.

In general, you shouldn't modify this file, or try to generate it yourself. Rather, let one of the supplied programs create and maintain it for you. There are two exceptions to this: if you wish to remove one guide from the index file, you can do this by hand with a text editor. Be sure to remove the file name and all nodes associated with it. The second case is when you must change the path or name of an

amigaguide file. This you can also do by hand. In this case, be sure that you enter the full path name. If you need to change many path names, it may be simpler to simply regenerate the entire index. However, a "search-and-replace" from your text editor will also do the job.

If **HyperHelp_cx** has any problems with an index file that you've modified, the easiest solution is to simply regenerate the index file with MakeIndex or **HyperHelp**.

In addition to the index file is the index guide. This file (which has the same name as the index file with a "_gd" appended) contains a list of all the guides the HyperHelp system has access to, organized into directories. This greatly speeds up the access times on some searches, provides almost immediate response to the ForceKey forcekey}, and provides a more structured view of the system. It is suggested that you not have all of your guides in the same directory, but that you divide them up into sub-directories which represent the domain of the group of files therein. For example, I have all of the RCRM Autodocs (converted into AmigaGuide format, of course) in a directory called "Eris:RKM-Autodocs/" and all of my MUI Autodocs in "Eris:MUI-Autodocs". You will find that if you place all of your guides in one directory that it will soon become difficult for you to find particular guides in the index.

1.23 The commodities program

Using HyperHelp_cx

To use HyperHelp_cx, position the cursor over a keyword in your text editor and press the standard hotkey (rshift-help). HyperHelp_cx will attempt to get the word from your editor and will search through the index for a similar word. If there is more than one match, HyperHelp_cx will display all of these matches. If no match is found (and the HyperHelp system is fairly liberal about what a "match" is), you will be presented with an index of all files HyperHelp has access to. See the section on the hotkey **editing** window option to control the appearance of the editing window when you press the hotkey.

A second way to use the commodities program is to be typing in a shell when you press the hotkey. HyperHelp_cx will get the last few characters you have entered and use them as the search muster (see

WatchDog). HyperHelp_cx is reasonably intelligent in that it will parse some editing commands for you; that is, it will understand simple editing keys such as the right and left arrows, delete, and backspace. If it encounters a character it does not understand (or that is ambiguous) it will consider that key to be the beginning of a new word. This method will always draw the muster from the last few keystrokes you've typed, whether you typed them in a shell, a string gadget, or an editor. Note that HyperHelp_cx will always request a keyword from the editor first through the ARexx port, if the editor's window is active. The third way to use HyperHelp_cx is to press the Force-hotkey combination (default ralt-help). This forces HyperHelp_cx to open an index of all files it has access to. With this method, no search muster is requested from the system. See the section on the forcekey **editing** window option for controlling the appearance of the editing window when the forcekey is pressed.

HyperHelp_cx waits in the background for you to press a hotkey. When this event happens, HyperHelp_cx will search through an **index** file (which you create with the help of a supporting file) for a keyword and opens an AmigaGuide file with information about that keyword.

Wildcards (?*) are supported in the keyword:

'?' - is a joker for exactly one character

'*' - is a joker for any number of characters.

Summary:

If the current active window does not belong to your text editor, HyperHelp_cx will look at the last few keystrokes you've typed and use these as the keyword. If the active window does belong to your text editor, HyperHelp_cx will attempt to request a word from the editor to use as the keyword. If both of these attempts fail, or if you press the ForceKey, HyperHelp_cx will open a window with a list of all the guides it has in its index. Depending on the settings of the editing window options, HyperHelp_cx may or may not open a small window to allow you to enter or edit a keyword.

1.24 Rights and reservations

The HyperHelp system (this includes all files in the distribution) is copyrighted 1994 by Apocalypse Productions.

This archive is freely distributable. All files except the sourcecode

archive must be present in any distribution of the package. No more than a reasonable copying fee may be charged for its distribution. The authors reserve all rights to the source code and executable programs. Neither the programmers nor Apocalypse Productions is liable to any damage that the use of this package may cause. There is no guarantee that this software will function on any computer.

I will not be held responsible if HyperHelp reformats your harddrive or blows up your monitor, or if the very existence of HyperHelp on your HD causes your friends to avoid you. I am not liable if the Mafia come and snuff you while HyperHelp is running, or if you use HyperHelp to help you break into National Security Council network or rob a bank. Since HyperHelp only runs under OS2.0 or higher, you can't blame me if you get an inferiority complex about having an earlier version of the OS and spend the rest of your life in a mental institution.

But if any of this stuff happens, let me know so I can brag about it in the next version.

This package is sort-of-shareware. There is nothing crippled in these programs, and we do not require that you send us money. I realize that I might as well say "don't send us anything," but I've always found it silly to demand something when you can't enforce it. This coupled with the fact that any cripple-ware that comes across my WorkBench goes straight into the Trashcan requires that we trust you, the users, to send us a contribution if you like and use the system. Nobody can make a living off of shareware, and it's silly to try... but we all like presents =-)

If you do decide to contribute, send the contribution (whatever it may be) to the programmer nearest you (Sean if you're in the Americas, Helmut if you're in Europe).

It will be interesting to see just how many of you shareware authors (especially of "crippled" software) make a contribution...

Which brings us to the mandatory plug:

This program is SHAREWARE! If you use this program, you must send me \$6000, so that I can buy a new Amiga 6000/060, when they come out. If you don't, the ETHICS POLICE will come to take you away!!

(That's a joke, Bob.)

REAL programmers don't charge shareware fees.

REAL programmers program for the sheer joy of programming, not from some inbred capitalistic instinct.

Whenever you write a program of worth and give it away for FREE (you

may, of course, retain your rights to the code), you get Karma points, which you can then use to buy your way into Programmer Heaven, where laptop Crays are free, and the television frequency wavelengths are occupied by a no-charge ICDN network.

IF you charge shareware fees, you get sent to The Bad Place, where you are forced to work round the clock on used Sinclair PCs with pressure keypads, programming accounting packages and missile guidance systems.

1.25 The CLI index creator

In the case that you don't have MUI, or are simply a fanatical CLI user, a program for generating **index** files is included in the HyperHelp package. This program is called MakeIndex, and takes a number of arguments which is uses to generate an index file.

If you run MakeIndex with no arguments you will get the following:

USAGE: makeindex <index> {files}

If you supply a question mark you will receive the standard template:

I=INDEX/A,F=FILES/M:

INDEX - This parameter is required. The INDEX is a complete path and file name of the index file you are using. If this file does not exist, it will be created. If the file does exist, the additions will be appended. Only one index file can be used by the commodities program at a time.

FILES - You may supply the names of the guides you wish to add to the index. If you do not supply this parameter, MakeIndex will open an ASL filerequester for you to choose your amigaguides with.

1.26 The CLI preferences program

You'll probably only be using this program if you find yourself using MakeIndex as well. This program sets the preferences file which simplifies using HyperHelp_cx. You do not need to use this program if you have a registered copy of **MUI**. The program HyperHelp provides a much more intuitive interface to do this job. Run MakePrefs with no options to get a description of the argument template, and read the section on the **HyperHelp** program to find out what all of the options mean.

This program will generate a preferences file in ENV: based on the information you give it. Note that if you use this program several things will happen:

- 1 - Any prefs file you previously had will be completely written over. Not only that, but it will be deleted first, denounced as a heretic, called bad names and then be written over.
- 2 - HyperHelp_cx will use those preferences in favor of CLI and icon arguments.
- 3 - Using this program will set all of the preferences, not just the ones you specify. Arguments that you do not specify will be written with the defaults.

The only argument you are not allowed to not enter is the **index**. This must be the full path and file name.

All of the other arguments have defaults.

Note that you must copy this file from ENV: to ENVARC: yourself, if you wish your prefs to be permanent.

EG:

```
1> copy env:hyperhelp.cfg envarc:
```

1.27 Contacting the programmers

Almost more than getting money from you, we would like to hear what you think about HyperHelp. Comments, suggestions, raving compliments... whatever. Tell us about your dog, if you like. (Well, tell Helmut about your dog...)

Send us bug reports!

Incompatibilities with your system, mysterious crashes, memory loss... whatever. If you think you've found a bug, you're probably imagining it, but tell us about it anyway =-).

The ideal way to contact us is, of course, through E-Mail. Although I (Sean) don't know what my address will be next year (yet), all EMail that you send to Helmut will be forwarded to me. We accept mail in English, German, Spanish, French, Danish, and Swahili. We will only reply to the first four, but the other two we could hang on our walls in joyous incomprehension.

Sean E. Russell -

At this moment, we don't know how Sean Russell can be contacted. After overstaying his welcome in Germany, he's finally going back to the States to finish his last year of computer science. Until then, he has

neither Net access nor a permanent address. Although the response delay may be a bit longer, mail can be sent to the following address:

Sean Russell

2906 NW Audene Drive

Corvallis, OR 97330

USA

Helmut Lutzenberger -

Helmut can be contacted at the following node:

lutzenbe@informatik.tu-muenchen.de

Helmut is also in the process of changing his physical residence, and can therefore not be directly contacted. However, mail can be routed through:

Helmut Lutzenberger

Bunsenstr. 24

81735 Muenchen

Germany

1.28 Bugs

The HyperHelp system was developed and tested simultaneously on the following systems:

* Amiga 4000/EC030

4MB RAM

80MB AT/IDE hardrive

Maxon C++ 1.10.3 // SAS/C

GoldED 0.98

* Amiga 2000

68030 Turboboard

8MB ram

200MB ALF3/SCSI harddrive

Picasso II graphiccard

SASC version 6.51

CygnusED version 2.11

BugList and Future

(b) represents a bug

(i) represents a desired improvement

(u) represents an undesirable feature (not a bug, but almost), or an improvement that is almost required.

(i) Saving/Using from HyperHelp should update HyperHelp_cx

(i) Optional "load index into memory" feature.

(i) A completely GUI (~MUI) oriented Preferences program for those unfortunates who don't have MUI.

(i) Users should be able get a list of all guides currently in the index from HyperHelp.

(i) A "remove" button in the HyperHelp program for forcing HyperHelp_cx to quit.

(u) Clipboard support

(i) _cx control through ARexx

(i) Links to normal ASCII texts in the index

(i) Locale library support (Almost superfluous in these programs)

During the course of this project, we discovered a multitude of bugs in code generated by the Maxon compiler, which caused several delays in development. In the final phase I aquired the SAS/C package, and developement and debugging went much more quickly. In addition, Maxon doesn't hold to the ANSI standard too well. It's unfortunate, because MCPPC a spicy little compiler. As it stands, MaxonC++ will not likely be used in any future Apocalypse products (including releases of HyperHelp).

1.29 General Q&A, and paradigms

We'll try to anticipate some questions you may have, and let you know why we implemented some things the way we did.

(Q) Why are all of the nodes in the index file upper-case?

(A) This is to speed up the search routine. The speed gain (~12%) wasn't quite what we expected (~25%), so if you'd really like to have your nodes displayed with the proper case, drop us a line. If we recieve enough requests we'll change it back.

(Q) Sometimes HyperHelp_cx doesn't work. Why?!?

(A) First, the only solution to this is to reboot. This is a bug in the AmigaGuide library, not in HyperHelp_cx. Please do not write us about this.

(Q) I've noticed that HyperHelp causes memory loss.

(A) So have we. 184 bytes per window that it opens, to be exact. This is a bug in the MUI library (version 7.973), not in HyperHelp. The author of MUI is aware of the problem and is working on it.

1.30 About Magic User Interface

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 "muiXXusr.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-Stra ße 7

80935 München

GERMANY

1.31 Credits

We'd like to profere thanks to the following individuals:

Asher Royce - For proofreading and distraction. Asher is a tremendous help to developers because he's one of those people who cause computers to crash by simply being in the same room as them. Since programmers unconsciously avoid bugs in their own programs, people like Asher are invaluable for the debugging process.

Karin Niggel - For being there, and rather extreme methods of distraction.

Olaf (Anrufbeantworter) Lange - For the harddrive (and for much Amiga support)

1.32 You did WHAT?!?!?

It's recently (04-May-94) been announced in Germay that Commodore USA has been bought out by some random consortium.

Normally we would say that this is Not Good. However, as long as the OS stays the same (with, er, minor improvements such as RTG and memory

protection allowed), this could be a move for the better. Certainly there exists no company in the world with a worse marketing strategy than Commodore ... except for Atari; but it was run by people who got kicked out of Commodore, so what do you expect?

In any case, assume for a moment the worst: This is the End.

No more support. (Did we ever really have any???)

This is the best time to really start supporting those shareware programmers. Already, the majority of good software out there is share/gift/freeware or PD. Due to the recent Bad News, this situation may become even more extreme. Since the people who are producing (some of) the best software for your computer, it would be wise if you supported them.

This goes double for us Americans.

Did you know that Americans are the single worst source of shareware contributors, world-wide? We have over three times the amount of people in the States than are in Germany (250 million vs. 80 million), and many shareware producers can count 10 times as many contributors from Germany than the States. We're well renowned as the cheapest users in the world.

I'm not making a plug for you to send money to us. It'd be nice, but we program for the love of programming, and as long as Amiga OS exists, we'll be making software for it. But there are many damned good programmers out there producing damned good software who need encouragement.

Invest in shareware... it's the future.
