Guidemaster

The *ultimate* Amigaguide viewer.

Copyright Notice

Disclaimer

This product is offered "as is", and neither the author nor Greywire designs will be held responsible for any damages incurred as the result of running this software.

Distribution

You may not make a profit from distributing Guidemaster. You may not put it in a disk collection without prior permission from the author. Specific exceptions are as follows: Any collection from Fred Fish, any official archive of the Aminet collection.

You may not modify or eliminate files from the archive; you may split the archive if convenient so long as all the files are still available on the distribution medium.

Certain parts of the package may be distributed separately, subject to agreement with the following requirements:

The "Guidemaster" main program may be included in any disk collection, or with any product that is freely distributable, for use as the viewer of (compressed) Amigaguide files. You must include somewhere in your documentation the following copyright notice:

Guidemaster. © 1995 Greywire designs. All Rights Reserved. Reproduced and distributed under license from Greywire designs.

If you are a disk distributer, or are distributing a commercial package, please contact the author and arrange to receive a license and the latest version.

The shareware registration fee is $10. You will recieve the latest version on disk, and possibly a printed manual if there is any demand for it.

About

This software is developed on an Amiga 500, with an 85MB hd and 4.5MB ram. The documentation is created with a combination of Makeinfo and TEX / Metafont. Printed manuals, if requested, will be printed on a Canon BJ-200 printer.

1 Author

This software is Shareware. The suggested registration is $10. You will recieve major updates on disk via mail, if requested.

Please report any bugs or suggestions to me, Aric Caley.

Internet: dances@qedbbs.com

NOTE: I can only receive mail, I CANNOT respond to messages from this
account! I can't afford to pay for a full account since the system went
commercial; hopefully if I get some registrations for Guidemaster I will be
able to. Nevertheless, please report bugs or suggestions, and I will try
to get back to you through other channels.

Snail mail (US Postal service):

8720 Valley View, apt J5,
Buena Park, CA. 90620.

Voice phone: (714) 229-9957

2 About

2.1 Introduction

This is, in my opinion, the first usefull version of Guidemaster; that is, it has the major features I require: speed, low memory usage, and linking. It still does not have all the "nice touch" features I want to add, mostly because it has become quite a wrestling match with Amigaguide to get any of this to work at all, and some things may be simply impossible to add without completely rewriting Amigaguide. I've had a royal bitch of a time trying to get this all to work properly, and the Amigaguide / datatypes system has fought me tooth and nail every inch of the way. But it is working. Some of the major problems that still exist:

- If individual nodes in an Amigaguide document use commands like "FONT", "TOC", etc – IE, any global command, they simply do not work. This is a problem with Amigaguide and Nodehosts, NOT Guidemaster!
- For some reason, on both an A4000 and A1200 running 3.0, Guidemaster just behaves erraticly when viewing Amigaguide files – scrolling doesn't work. Other datatypes work just fine though! It works on my 3.1 system. I have no idea what the problem is. Documenation for Datatypes is poor and gives me no clue as to what's wrong. I've traced the problem to the `datatypes.library` function `SetDTAttrs()`. I try to set the attribute `DTA_TopVert` to match the value of the scroll bar. In 3.1, this causes the datatype to scroll and refresh; in 3.0 it refreshes but never scrolls. If anybody knows what I'm doing wrong, please let me know.
- Linking now works but only in 3.x. I haven't been able to get my custom Datatype to work, so I have found a solution that works with the dynamic node host. For no particular reason at all, Amigaguide behaves differently under 1.3 and 2.x, such that my linking technique doesn't work (it SHOULD, if Amigaguide just behaved the same as in 3.x). The only other way to make it work in 1.3/2.x is to make a patch to DOS Open() and wait for the Amigaguide task to open a compressed file. Ugh! I'd rather not do this, but if there's a demand for it, I will.

2.2 General

Guidemaster solves several problems with the current Amigaguide (1.3-2.1) and Multiview (3.0-3.1) viewers. First and foremost, you could not view compressed files. Since Amigaguide files compress very compactly, and since they are often quite large, it is of great advantage to store them in compressed form. Another anoying problem was the apparent loss of some keyboard shortcuts under AmigaOS 3.0/3.1. Then there are the speed problems – on my poor little A500, it sometimes took many long frustrating seconds for Amigaguide to pop up when reading a large help file.

Guidemaster lets you easily view compressed Amigaguide files (under 3.0/3.1, it even displays any file which you have a Datatypes driver for). Using XPK, you have the choice of many types of compression, whatever suits your needs. But unlike other solutions, Guidemaster does not need to load the whole file and uncompress it in ram all at once. Other programs must do this, and it wastes a lot of memory (not to mention being slower) – a 100K file might uncompress to 300K, and then Amigaguide itself then loads a good portion of that into memory as it displays the file (that 100K file might explode to consume 500K or more of ram!).

To conserve on your ram, and to make things faster, Guidemaster uses a custom format for compressed Amigaguide files. This format allows Guidemaster to load small chunks and uncompress them only when needed. Upon initialy viewing the file, Guidemaster reads just a few small chunks and then immediately displays the first node – thus, it opens up its window faster than Amigaguide *normaly* does! Amigaguide scans through the file first, Guidemaster doesn't need to do this because the custom format file includes an index. Included is a program to convert and compress Amigaguide files into the custom compressed format (Packguide).

Of course, you can still view normal Amigaguide files and Amigaguide files compressed the old way – but it will be slower and use more ram. It's your choice. Under 3.0 and above, you can also view any file that has been XPK compressed and you have a Datatypes driver for.

2.3 Features

- Linking between compressed files (3.x only)
- Keyboard shortcuts for all actions.
- Will display:
  - Normal Amigaguide files compressed with xPack/xPk/etc (slower, memory hog).
  - Custom IFF AGID format, uses XPK internaly. Faster and more efficient than normal Amigaguide!
  - Any datatype object that has been XPK'd.

2.4 Future

These are some of the features I plan for the future:

- AppIcon for viewing files. You can do this with Toolmanager anyway...
- Help directory node (lists all documents in the directories listed in ENV:Amigaguide/path) from the nodehost.
- Commodities hotkey brings up the help directory.
- Index of all nodes in document.
- Working search (node, document, all documents).
- Working keyword index search, index dynamic node from host.
- Multiple bookmarks, ability to save/load default bookmarks and collections of bookmarks.
- ARexx port.
- Amigaguide preferences: Default font, paths, Default colours.

Considering the trouble I experienced getting this to work, I would like to rewrite the Amigaguide datatype and Amigaguide library. In the process I would fix all the problems, plus add new features like the ability to embed any datatype object – and have text flow around it. I would like to do this first as a new text datatype, a hypertext subclass and a new Amigaguide subclass of that. I would also like to have an HTML subclass.

3 Installation

Just copy Guidemaster and Packguide to wherever you put your favorite utilities. Guidemaster will work in 1.3, 2.x or 3.1. Packguide needs 2.0 or above (I made Guidemaster 1.3 compatible so that anybody can view files. But I really don't care much to support 1.3)

You also need to modify the file "ENV:Amigaguide/path". You must add a set of double quotes ("") to the path list. This tells Amigaguide to also check in the current directory for a file, which apparently it doesn't normaly do (instead, it checks in the directory the current Amigaguide file came from). This is potentialy usefull even if you aren't useing Guidemaster. If you don't do this, Amigaguide will have a hard time locating some documents.

4 Usage

There are two programs in the Guidemaster "suite". There is the Guidemaster program itself, which is a replacement for the Amigaguide comand under 1.3/2.x and Multiview under 3.x. It can view any Datatypes object under 3.x, and can view normal Amigaguide files as well as compressed Amigaguide files.

In order to recieve the full benefits of the Guidemaster system, you must compress your Amigaguide files with the Packguide program! Packguide compressed files use XPK, and store the file in many smaller chunks. Using an index, Guidemaster is able to load and uncompress only the chunks you want to look at – thus, it is faster and uses less memory than programs like XPKGuide or even normal Amigaguide. You may wonder how small the chunks are – they are typicaly around 32K, which I have found to be a good compromise between unpack speed, memory useage and compression efficiency.

4.1 Packing

For a description of the Format and Template lines used below, see Section 6.2 [Formats and Templates], page 10.

To compress an Amigaguide file, use the Packguide program. Packguide requires 2.0 or better at this time, sorry. This is the template:

```
FILE/A/M,S=SUFFIX/K,M=METHOD/K,ALL/S,QUIET/S
```

`Format: FILE <name>`
`Template: FILE/A/M`

The Amigaguide file(s) to compress. Wildcards and multiple file names are accepted.

`Format: S=SUFFIX <name>`
`Template: S=SUFFIX/K`

An extension to put on the resulting compressed file. If you don't specify this option, the compressed file will replace the original!

** ALWAYS KEEP BACKUPS OF YOUR AMIGAGUIDE FILES! **

`Format: M=METHOD <type>`
`Template: M=METHOD/K`

The XPK compression method to use. NUKE is the default.

`Format: ALL`

`Template: ALL/S`

If input defines any directories, the `ALL` keyword tells Packguide to go into each directory and compress all the files therein.

`Format: QUIET`
`Template: QUIET/S`

Don't print any information while compressing.

4.2 Viewing

To view a file, use Guidemaster, which works in 1.3 and up:

```
NAME,S=PUBSCREEN/K
```

`Format: NAME <name>`
`Template: NAME`

Amigaguide or AGID file to view (or any Datatypes supported format under 3.1). If no name is given, a file requester will ask for a file.

`Format: S=PUBSCREEN <name>`
`Template: S=PUBSCREEN/K`

Name of a public screen to open on. The name is case sensitive.

5 Wish list

There are many things about Amigaguide that annoy me to no end:

Why don't "ALINK", "CLOSE", and "QUIT" work anymore?

Why can't you put anything you want in a macro? Only attributes will work.

Why isn't "FONT" an attribute so I can change font mid-line?

For that matter, is there any particular reason why most commands have to start at the first column in a line?

Why does a double backslash make a single backslash only in V40, and a single backslash doesn't show at all? This makes it impossible to use backslashes in your text and have it show properly on all Amigas.

Node Hosts:

Why can't I bind a node host hook to a single document?

Why does Amigaguide scan the list of node hosts even when it is possible to send the packet directly to the right host?

Why, in a node host, the node returned cannot use many commands (they just don't work)? Things like TOC, PREV, NEXT, FONT, etc simply don't work. ARGH!!!

Why can't I return the name of a file and have Amigaguide display it? For some reason, this "feature" simply doesn't work (Amigaguide DOES open the file, you just don't SEE it!!).

Why doesn't the Amigaguide Datatype let you disable the gadget bar? Why isn't it a subclass of the text datatype? Why do all the datatypes drivers, well, suck in general?

Why doesn't it (specificaly the Amigaguide DT) let you "get under the hood"?:

I want to be able to have a render hook.

I want an unknown command hook, so I can add new commands.

I want a layout hook so I can cause text to flow around graphics.

I want a node host hook that is local to one Amigaguide file.

Datatypes in general:

DTM_SELECT appears to not work, period. !#*%& This prevents me from doing proper cut and paste operations!

Picture datatypes:

Consume chip ram like there's no tommarow (or like there's a tommarow with 8 MB chip ram...)

Don't support 24 bit.

Are slow.

Make little or no attempt to scale the image so the aspect is correct, or do decent dithering, etc.

6 Formats and Templates

6.1 Format

Commands are described with a Format and a Template. In a Format specification, the arguments are surounded by brackets to indicate the type of argument. The brackets are not typed as part of the command.

`< >` Angle brackets are used to indicate that this argument is *required*; it must be provided or the command will fail.

`[ ]` Square brackets indicate that the argument is optional. The command will run with or without these arguments.

`{ }` Braces indicate that this argument may be given more than once.

`|` The vertical bar separates multiple options, only one of which may be specified (mutualy exclusive).

6.2 Template

Templates are a more compact, concise version of Formats (both have their uses). Templates are directly supported by the Amiga OS. Their main advantage is that they specify the type of data each argument will (should) be. Each argument is separated with a comma, and has a specifier code at the end (always a '/' with a letter). These are the codes currently supported:

`/A` Always required. This argument *must* be given or the command will fail. This is equivalent to the Format specifier `"< >"`.

`/F` Final argument. The entire rest of the line, regardless of any keywords or spaces that may appear in it, is taken as the argument string.

`/K` Keyword. This option will only be filled if this keyword appears in the command line.

`/M` Multiple arguments. This argument will accept any number of strings. Anything not matching another option will be added to this option. Only one /M will be specified. If there are any /A arguments after the /M argument that are unfilled, strings will be taken from the end of the the /M argument.

`/N` Number. The argument is a decimal number.

`/S` Switch. The argument is a boolean switch. If it is specified, the option is true, if it is missing, the option is false (default).

`/T` Toggle. Just like a switch (/S), but causes the boolean value to toggle.

`=` This is used to provide an abbreviation. `OPT=OPTION` means that this option can be specified with either `OPT` or `OPTION`.

If for some reason you need to specify an argument string (for instance, a file name) that is the same as one of the options, enclose it in quotes.

For example:

Hypothetical command `Show` with the Template: `NAME/A`, and you have a file called `"name"`.

You would use:

```
show name "name"
```

Table of Contents