Amigaguide

Documentation for Amigaguide commands through V40 AmigaOS.

Written by Aric R. Caley. Freely distributable.

About

This document was written on an Amiga 500, with an 85MB hd and 4.5MB ram. The documentation is created with a combination of Makeinfo and TeX / Metafont.

1 Amigaguide

1.1 About

Like other people, I did not have a single concise source for all the Amigaguide commands and how they work (or don't as the case may be). So I decided to write one. I believe I have covered all the commands from the earliest Amigaguide to the latest (release 3.1 of AmigaOS).

I've also considered writting more about how to work with Amigaguide as a programer (how to work with dynamic hosts, the Amigaguide Datatype, etc), but that will only come later if there is a demand. I also thought about doing a general tutorial on writting Amigaguide documents (or Texinfo documents for that matter).

I don't claim to be any expert on Amigaguide, but, I do think I do a fairly good job at rounding up information and compiling it into a nice readable form. I hope you will agree.

Aric R. Caley.

Internet: dances@qedbbs.com

Snail mail (US Postal service):

5829 Los Feliz,
Buena Park, CA. 90620.

Voice phone: (714) 229-9957

1.2 Background

Amigaguide commands all begin with the '@' (pronounced 'at') symbol. Most commands must start at the very begining of a line, and should be the only thing that is on that line. Although not required for most commands, you should probably write them all in capital letters as a matter of discipline (it also makes them much easier to see and keep track of!).

There are three catagories of commands: Global, Node, and Attributes. Global commands are usualy specified at the begining of the document, before any nodes are defined, and apply to all the nodes in the document. Technicaly, they could be anywhere. Many commands can be used both globaly and in nodes.

Node commands are usable inside of a node (after a '@NODE' and before an "@ENDNODE"), and affect only the node in which they are used.

Attributes may be specified anywhere in a normal line. In addition to the '@' symbol, attributes always use a pair of braces ('{' and '}') to enclose the attribute name and possibly additional arguments.

Many commands have been introduced over time, so some commands require a minimum OS version. These commands may be used without too much worry about compatibility since older versions of Amigaguide largely ignore unknown commands. Where a command only works on more recent versions of the OS, they will be followed by a "V39" or "V40", otherwise they work on any Amiga. V39 is AmigaOS/Workbench release 3.0 (only on A1200's and A4000's), and V40 is release 3.1 (available to anyone willing to pay through the butt for it).

To avoid complications in viewing this document with Amigaguide, all the commands listed will not include the necessary '@' – but don't forget it when you are authoring an Amigaguide document!

1.3 Paths and Nodes

The commands INDEX, HELP, NEXT, PREV, TOC and the all purpose hypertext link specify other nodes to jump to. They all support the naming of nodes within the current document, but they also all support a path along with that name which lets the node be located in any Amigaguide document.

It is really the essence of the simple. To access a node in another document, simply put an AmigaDOS file path before the node name. You do not need a complete path including a volume name, if you are sure that the file is going to be where it should be (generaly, if it is not in the same directory as the main document, you should use a full path, maybe an assigned volume name). Some example paths are: "s:help.guide/main", "Amigaguide.guide/Copyright".

If you are running AmigaOS 3.0 or above, you can jump to any file that is supported by Datatypes – pictures, animation, anything. You must still give a "node" name, even though the file is not an Amigaguide file and has no "nodes" – so just use "main": "picture.iff/main".

2 Global Commands

2.1 Global Commands

`$VER: <AmigaDOS version string>`

Specify an Amiga standard version string. "VER" must be uppercase. The format for an Amiga version string is: `<name>` (mm/dd/yy). If you have this in your Amigaguide file, you can use the AmigaDOS command "Version" to get the string.

`(C) <copyright>`

Specify a copyright notice for this file.

`AUTHOR <name>`

Specify the author of this file.

`DATABASE <name>`

Identifies this file as an Amigaguide document. Must be the very first thing in the file!

`DNODE <name>`

This is an obsolete command, do not use. It used to be for indicating a "dynamic node", but is not needed because the Amigaguide system will search for a node that doesn't exist, at which point a dynamic node host can return the node.

`ENDNODE`

Specify the end of a hypertext node definition.

`FONT <name> <size>`

Specify the font to use for the database.

`HEIGHT <rows>`

The number of rows in the largest node.

`HELP <path/node>`

Specify a help node accessible with the "Help" button.

`INDEX <path/node>`

Specify an index node accessible with the "Index" button.

`MACRO <name> <commands>`

Define a macro for this document. A macro is made up of attribute commands and or other macros, and is used like other attributes. A macro can override the built in attributes. A macro can accept arguments by embeding a '$' symbol with a number, for each argument supported; IE, '$1', '$2', etc. If an argument is given when using the macro, but is not defined in the macro, then it simply doesn't appear, which might be usefull to somebody. V40.

Example:

```
@macro bold "@{b} $1 @{ub}"
  ...

This is an example of a @{bold "bold"} macro.
```

MASTER <path>

Specifies the AmigaDOS file path of the original document this Amigaguide file was derived from.

NODE <name> <title>

Specify the beginning of a hypertext node definition.

ONCLOSE <rxcommand>

Specify an ARexx script to execute when this document is closed. The script's return value is not significant. V40.

ONOPEN <rxcommand>

Specify an ARexx script to execute when this document is opened. The document will not be opened if the script returns an error. V40.

REM <remark>
REMARK <remark>

Used to place a remark in the document which is not displayed.

SMARTWRAP

A more inteligent word wrapping algorithm than WORDWRAP. Paragraphs must be separated by two line feeds (visualy, a blank line). V40.

TAB <num>

Define the size in spaces of a tab. Defaults to 8. V40.

WIDTH <chars>

The number of columns in the largest node.

WORDWRAP

Use wordwrapping for all paragraphs. To be wordwrapped, the paragraph must have no line feeds. V39. Since this is ugly to work with, and ugly to look at in Amigaguide 2.x and lower, you should use SMARTWRAP and write your Amigaguide files to look wordwrapped – this makes your document look decent on all systems while taking advantage of V40 and above since SMARTWRAP will be ignored on pre-V40 Amiga's.

3 Node Commands

3.1 Node Commands

\
       Sometimes it is necessary to be able to use the '@' symbol in your document. To prevent it from being interpreted as a command, use the '\' to escape it: '\@'. To use a '\' itself, you must also escape it: '\\'.

FONT <name> <size>
       Specify the font to use for this node. The name must be the complete name of the font, IE: "times.font".

HELP <path/node>
       Specify a help node accessible with the "Help" button.

INDEX <path/node>
       Specify a help node accessible with the "Help" button.

KEYWORDS <keywords>
       Keywords for this node. Not currently used by Amigaguide.

MACRO <name> <commands>
       Same as the global version.

NEXT <path/node>
       Define the next logical node (if not physicaly located right after this one) accessible with the "Browse >" button.

ONCLOSE <rxcommand>
       Specify an ARexx script to execute when this node is closed. The script's return value is not significant. V40.

ONOPEN <rxcommand>
       Specify an ARexx script to execute when this node is opened. The node will not be opened if the script returns an error. V40.

PREV <path/node>
       Define the previous logical node (if not physicaly located right before this one) accessible with the "< Browse" button.

SMARTWRAP
       Same as the global version, but only effective for this node. Note: No matter where SMARTWRAP appears in the node, it will effect the *entire* node.

TAB <num>
       Define the size in spaces of a tab. Defaults to 8. V40.

`TITLE <title>`

> Specifies the title of this node, which is displayed in the title bar of the window when this node is selected. Must begin at the start of a line.

`TOC <path/name>`

> Specifies the table of contents for this node (the default is the node "MAIN") accessible with the "Contents" button.

`WORDWRAP`

> Same as the global version, but only effective for this node.

4 Attribute Commands

4.1 Attribute Commands

`{<label> <command>}`

Specify a hypertext link. Not really an attribute per se, but shares a similar syntax and scope. It may be specified anywhere on a line.

The command is usualy "LINK", to specify a normal hypertext link. There are other commands as follows:

`ALINK <path/name> <line>`

Display the hypertext node in a new window, starting at `<line>`. This no longer works in V39 and up (why not??).

`CLOSE`   Close the window, to be used with ALINK windows. Doesn't work in V39 and up.

`LINK <path/name> <line>`

Display the hypertext node, starting at `<line>`.

`RX <command>`

Execute an ARexx script.

`RXS <command>`

Execute an ARexx string file.

`SYSTEM <command>`

Execute an AmigaDOS command.

`QUIT`   Shutdown the current database. Doesn't work in V39 and up.

The node is the name of a node in this document, or a path to a node in another document. In AmigaOS 3.0 and above, the node may be a path to any type of file which is recognized by the Datatypes system.

`{AMIGAGUIDE}`

Displays the word "Amigaguide(R)" in bold. V40.

`{APEN <n>}`

Change the foreground (text) colour to specified pen number. V40.

`{B}`

Turn on bolding. V39.

`{BG <colour>}`

Change the background to a preferences defined colour. Colour can be:

`Text`

```
            Shine

            Shadow

            Fill

            FillText

            Background
            Highlight
```

{BODY}

Restore default formatting for normal body text. V40.

{BPEN <n>}

Change the background colour to specified pen number. V40.

{CLEARTABS}

Restore default tabs. V40.

{CODE}

Turn off wordwrapping. V40.

{FG <colour>}

Change the foreground colour. See command BG for colours. V40.

{I}

Turn on italics. V39.

{JCENTER}

Turn on centered justification. V40.

{JLEFT}

Turn on left justification. V40.

{JRIGHT}

Turn on right justification. V40.

{LINDENT <n>}

Specify an indent in spaces for the body of paragraphs. V40.

{LINE}

Force a line feed without starting a new paragraph. V40.

{PAR}

Specifies the end of the paragraph, equivalent to two line feeds and usable with SMARTWRAP. V40.

{PARD}

Reset to default paragraph settings: APEN to 1, BPEN to 0, original font, and LINDENT to 0. V40.

`{PARI <n>}`

> Specify an indent in spaces for the first line of a paragraph. Value is relative to LINDENT and may be negative. V40.

`{PLAIN}`

> Turns off all style attributes (bold, italics, underlineing). V40.

`{SETTABS <n> ... <n>}`

> Specify a series of tab stops in spaces. V40.

`{TAB}`

> Outputs a real tab character. V40.

`{U}`

> Turn on underlining. V39.

`{UB}`

> Turn off bolding. V39.

`{UI}`

> Turn off italics. V39.

`{UU}`

> Turn off underlining. V39.

Table of Contents