

Heddley

Edd Dumbill

COLLABORATORS

	<i>TITLE :</i> Heddley		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Edd Dumbill	July 1, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Heddley	1
1.1 Heddley v1.1 (C) Edd Dumbill 1994	1
1.2 Introduction to Heddley	2
1.3 About the author	3
1.4 Version information	5
1.5 Installing Heddley	9
1.6 Editing the database	10
1.7 Editing a document	11
1.8 Other Information	12
1.9 The Add document gadget	13
1.10 The Attrib gadget	13
1.11 The Delete gadget	14
1.12 Clear text	14
1.13 Load text	14
1.14 Edit text	14
1.15 Make into button	15
1.16 Delete button	15
1.17 The Button Edit window	15
1.18 Load database	16
1.19 Save	17
1.20 The Info Window	17
1.21 Starting a new database	18
1.22 The document name list	18
1.23 Starting Heddley	19
1.24 Configuring Heddley	19
1.25 The ARexx port	20
1.26 The FONT tooltype	21
1.27 The FONTSIZE tooltype	21
1.28 The EDITOR tooltype	22
1.29 The TEXTWIDTH tooltype	22

1.30	The HELPFILE tooltype	23
1.31	The FGPN tooltype	23
1.32	The BGPEN tooltype	23
1.33	ADD	23
1.34	ATTRIB	24
1.35	DELETE	24
1.36	EDIT	24
1.37	GOTO	25
1.38	INFO	25
1.39	LOAD	26
1.40	LOCK	26
1.41	NEW	27
1.42	PENS	27
1.43	QUERY	28
1.44	QUIT	30
1.45	SAVE	31
1.46	TEST	31
1.47	PREFS	32
1.48	MOVE	32
1.49	ARexx examples	32
1.50	Distribution of Heddley	33
1.51	Saving HTML source	34
1.52	The Preferences Window	35
1.53	The Palette Gadget	36
1.54	The Current Pen cucle gadget	36
1.55	The Font Button	36
1.56	Document Pick List	36
1.57	Test Database	37
1.58	On line help	37
1.59	AmigaGuide ARexx Port	37
1.60	Quick start guide to using Heddley	38
1.61	Things I plan to implement in Heddley	39
1.62	Moving a document - Up	39
1.63	Moving a document - Down	40
1.64	Change text styles	40
1.65	What About...?	40
1.66	AmigaGuide V39+	41
1.67	Why should I use Heddley?	41
1.68	How do I?	42

1.69 Replicate button	43
1.70 Smooth scroll	43
1.71 Save as ANSI	43
1.72 Writing Amigaguide files	44
1.73 Save as ASCII	47
1.74 Index	47

Chapter 1

Heddley

1.1 Heddley v1.1 (C) Edd Dumbill 1994

Heddley v1.1

(C) Edd Dumbill 1994

v1.18 release

HEDDLEY IS A SHAREWARE PRODUCT

READ THE DISTRIBUTION SECTION FOR INFORMATION ON DISTRIBUTION
AND REGISTERING

Contents

~IMPORTANT~---~DISTRIBUTION~
distributing and registering

~Feature~list~-----
why you should use Heddley!

~Introduction~-----
about Heddley

~Installing~Heddley~-----
installing Heddley on your system

~Starting~Heddley~-----
starting the program

~Configuring~Heddley~-----
setting your preferences

~Quick-start~guide~-----
for a very quick taste

~Editing~a~database~~~~~
detailed function explanation

~On-line~help~function~~~~~
getting help when you need it

~The~ARexx~port~~~~~
Heddley's ARexx support

~ARexx~example~scripts~~~~~
useful examples

~HTML~output~~~~~
get your databases on the WWW!

~How~do~I~do...~?~~~~~
how to do the non-obvious

~What~about...~?~~~~~
queries about Heddley

~Style~and~things~~~~~
guidelines on writing AmigaGuide databases

~Version~information~~~~~
what's new, fixed and changed

~To-do~feature~list~~~~~
what's left to do

~Other~information~~~~~
a miscellany

~How~to~contact~the~author~
for bug reports and registration

~Index~~~~~
find what you want...

An HTML version of this documentation, created with Heddley, is available on the WWW at: <http://www.york.ac.uk/~ejad1/heddley.guide/main.html>

The latest version of Heddley should be obtainable from the page: <http://www.york.ac.uk/~ejad1/amiga.html>

Both Heddley and this documentation are copyright (C) 1994 by Edmund Dumbill.

1.2 Introduction to Heddley

Why was Heddley written?

Heddley was written as there did not seem to be any decent programs around for editing AmigaGuide®~databases. Using a text editor became too odious when you kept forgetting the nawes of nodes etc; Heddley is designed to remove all the tedious tasks from creating AmigaGuide® databases, in order to enable you to concentrate on their content.

Heddley allows you to compose AmigaGuide® guide files with greater ease than by hand. It provides you with an overview of all the documents in the database (an AmigaGuide® file is called a 'database', made up of many documents). Now Heddley also allows you to save your database as HTML source, for use with AMosaic or some other WWW reader, and as plain text with ANSI style information. This means that Heddley should suffice for all your documentation needs.

Heddley works in conjunction with your favourite text editor, to provide a convenient interface for editing each document within the database. The documents in the database are arranged simply in a list, and so it is easy to see at a glance the composition of your documents. Adding buttons has never been easier either; all you have to do is highlight the text in the document and select the relevant menu option! Adding text styles is just as easy and the full set of

AmigaGuide®~V39
style controls can be used.

Heddley also has a comprehensive

ARexx~port

, enabling you to write scripts to

automatically create AmigaGuide® files, and interface Heddley more closely with your text editor. It also has a full

on-line~help

facility detailing

every program function.

My grateful thanks go to Geir Harald Hansen (geirhans@ifi.uio.no) for his invaluable help with debugging Heddley and suggestions as to how to improve it. Credit for suggestions of his are marked in this documentation with [ghh]. Also I want to thank Daniel Stenberg (Bagder) for his great help with debugging and suggestions. Where I've remembered, his input is marked with [bagder]. Look out too for his coming-soon text editor "FrexxEd". Like, frexx, man.

System Requirements

- * An Amiga running Kickstart 2.04 or greater.
- * amigaguide.library V34+ (in order to use the
Help
and
Test~Database
functions).

For best results a 68020+ (or accelerated 68000) CPU is needed - Heddley might be a little sluggish otherwise!

1.3 About the author

Heddley is (C) and was written by Edd Dumbill.

There are almost certainly bugs in Heddley; if you find one, or have a suggestion to improve Heddley, please contact me, giving as full and *precise* description as you can. See also the

'to-do'

section to see what I already plan to implement.

E-mail address:

ejad-a@minster.york.ac.uk

Valid at least until end of June 1995

IRC:

Edd (look for me on #amiga or #Heddley)

Snail-mail address:

Edd Dumbill,
13 Giles Avenue,
Burnholme,
York,
YO3 0RB.
England.

Telephone +44 (0)904 414524

Above valid until end of June 1995

WWW home page:

<http://www.york.ac.uk/~ejad1/>

Heddley on-line documentation:

<http://www.york.ac.uk/~ejad1/heddley.guide/main.html>

If you're lucky enough to have OS3.0+ and a GIF datatype, click~here to see a picture of me!

Thanks go to:

Wonder testers - Geir Harald Hansen
what stars!!! - Daniel Stenberg (Bagder)

Testers & - Anthony Moringello, Dave Hollway,
bug reporters - Olav Mueller, Carl Leduc, Stefan Eggers,
- Ralph Seichter, Lionel Vintenat, Dale Larson,
Jesper Skov.

First fee payer - David Pipe.

Neil Topham,

Ben Lidgley - for something or other.

God - for everything!

1.4 Version information

Version information

This is Heddley v1.18.

V1.18 - 29 August 1994 - SECOND PUBLIC RELEASE

V1.172beta - 26 August 1994

V1.172beta BUG FIXES

- * fixed document edit window not working when Workbench not running
- * hopefully fixed Enforcer hits after loading non-guide file in

V1.172beta IMPROVEMENTS

- * added form-feeds at page breaks to ASCII/ANSI save
- * use GTLV_MakeVisible in listviews in v39+

V1.171beta - 18 August 1994

V1.171beta BUG FIXES

- * fixed EDIT LOADTEXT for blank documents; refused to load new text before if document blank.
- * fixed horrible crash on ANSI save with no Autoindex.
- * hopefully fixed amigaguide.library v34 crashes when started from CLI. can't guarantee it though.

V1.171beta IMPROVEMENTS

- * recompiled with DICE inlined library calls. Should make executable smaller and hopefully a tad faster.
- * made progress indicator less flickery.
- * added STRIP option to SAVE ANSI which removes all ANSI codes and just saves plaintext. added save as ASCII menuitem.

V1.17beta - 16 August 1994

V1.17beta BUG FIXES

- * fixed save of prefs window & Enforcer hit in 'save prefs'
- * fixed test guide 'caching' problem
- * hopefully fixed Enf. hits in 'load new text'~in doc. window
It seems a compiler error caused this... :(
- * fixed bug which meant a document got an extra space on the end of it each time it was changed
- * fixed scroller in doc. edit window; now has proper NewLook

V1.17beta IMPROVEMENTS

- * added OK/CANCEL gadgets to most requesting windows
-

- * added 'Replicate button' menuitem in document edit window
- * made smooth scroll optional in document edit window
- * made front panel sizeable!!! I hope everyone appreciates the hard work (no MUI here thanks very much ;)
- * made slight speed improvement of text justify routine
- * tidied up source code (ugh did I write *this*? ;)
- * added progress indicator to loading and saving
- * implemented very basic 'save-as-ANSI' routine - ASCII with ANSI codes for text style. suggestions for improving this welcomed
- * added double-click word marking mode in document edit window
- * added caret pointer when marking text in document edit window
- * added style guidelines to documentation ;)

V1.16beta - 7 August 1994

V1.16beta BUG FIXES

- * fixed sizing behaviour of document edit window; with certain sized fonts the width changed each time you used it
- * fixed up scroller on Doc Edit window - should be resolution sensitive now.
- * fixed listview following
- * fixed AppWindow text loading; now checks to see it's not a directory

V1.16beta IMPROVEMENTS

- * added saving default position of windows into prefs file.
- * added support for @font and @wordwrap
- * improved memory allocation
- * made listviews follow current item; looks nicer
- * added page up/page down in document edit window
- * added user confirm on delete document

V1.15beta - 3 August 1994

V1.15beta BUG FIXES

- * fixed keycut indicator for font size in preferences window
- * fixed strange-last-char error with justification

V1.15beta IMPROVEMENTS

- * added text control output to HTML output.
- * improved HTML output to be clever and use proportional fonts.
- * altered behaviour of text control editing; now text control is absolute.
- * added Help, Author, Copyright & Version attribute handling in Info window.

V1.14beta - 25 July 1994

V1.14beta BUG FIXES

- * fixed remaining Enforcer hit in Document Edit Window [ghh]
 - * fixed 3.0 NewLook menus in Front Panel [ghh]
-

- * fixed some strange behaviour with spaces - now spaces inside buttons aren't messed with on justification.
- * fixed more space behaviour; means loading time after edit is doubled though. There must be a better way... ;-)

V1.14beta IMPROVEMENTS

- * rejigged button edit window and added Create Document option in there to make guide writing easier
- * added Up/Dn gadgets and shift-cursor keycuts & MOVE internal command to enable ordering of documents in list
- * jazzed up GUI appearance
- * >>> Added saving/loading/displaying of text control commands <<< I hope this makes people happy! Editing coming soon...
- * WOW! Put text control editing in too! Brilliant or what? :^)

V1.13beta - 17 July 1994

V1.13beta BUG FIXES

- * fixed quite a few memory leaks (thanks to Simon Bullon for his mem. debugging routines)
- * now loads guides where there are spaces between at-leftbrace and opening quote in buttons
- * fixed filerequest; now pattern gadget appears as well as pattern matching occurring (wanted to load a non-.guide suffixed file? :-)

V1.13beta IMPROVEMENTS

- * added on-line help facility
- * added 'Test Database' facility
- * added HELPFILE command line arg/tooltype

V1.12beta - 13 July 1994

V1.12beta BUG FIXES

- * fixed more Enforcer hits; in particular the NULL-document-title one (ugh - every time Edit is used). Thanks for all the reports guys...
- * fixed @prev in database output - duplicated the @next. Oops!

V1.12beta IMPROVEMENTS

- * drew Magic-WB style icon for Heddley

V1.11beta - 11 July 1994

V1.11beta BUG FIXES

- * fixed quite a few Enforcer hits (thanks Carl!); fixed hits when interpreting some internal commands & on testing for font loading on program init. There'll still be hits I think if syntactically shonky AG files are loaded in. Must fix this...

V1.11beta IMPROVEMENTS

V1.10beta - 10 July 1994

V1.10beta BUG FIXES

- * fixed bug in New Window gadget which meant that once you clicked it on, you couldn't turn it off again! Ooops! Sorry...
- * fixed keycut for Index choice in Index window. How did that one escape me? Anyway it works now.
- * fixed crash on attempt to load non-guide file.

V1.10beta IMPROVEMENTS

- * added automatic index creation feature. Nice :-)
- * added HTML saving option. Very nice :-)
- * made buttons use bevelbox instead of coloring - removed BUTTPEN tooltype and command line param.
- * added PREFS internal command (prefs. Window and saving prefs into tooltypes)
- * added menu strip to front panel window
- * added loading and saving of prefs.

V1.05 - 4 July 1994 - FIRST PUBLIC RELEASE

V1.05 BUG FIXES

- * fixed transposition of RX and RXS in Button Edit window
- * made document names fully case independent

V1.05 IMPROVEMENTS

- * done internal restructuring to start using an internal command based system; all commands of which are also accessible from the ARexx port
- * added facility for asynchronous document editing using ARexx port
- * added internal variables (attributes) which make some of Heddley's status available to the ARexx programmer
- * added handling of next and prev directives, and improved Attrib window to include them.
- * added Rexx scripts - also improved Install script
- * written gallons of documentation :*)
- * added handling of ALINKs via the New Wnd option in the button edit window

V1.04beta - 14 June 1994

V1.04beta BUG FIXES

- * fixed button type cycle gadget problem when used with Cycle2Menu
- * fixed highlight problem when shelling to editor
- * added document edit window center-on-open

V1.04beta IMPROVEMENTS

- * added tooltypes for foreground, background and button colors in document edit window
-

- * added ARexx port - we are cooking!

V1.03beta - 12 June 1994

V1.03beta BUG FIXES

V1.03beta IMPROVEMENTS

- * user-interface now fully font-adaptive
- * document edit window now uses font of user's choice
- * text output can be justified to any width of user's choice
- * added ToolType parsing for Workbench
- * added command line template parsing for CLI

V1.03beta - 9 June 1994

V1.02beta BUG FIXES

- * menus now use screen font rather than topaz 8.
- * screen title in button edit window fixed
- * multi-line buttons now simulated by repeated single line buttons
- * icon save now grabs ENVARC:SYS/def_guide.info properly if it exists. icons are not overwritten if they exist already.
- * will not crash if database with no Main node loaded - instead adds a Main node.
- * fixed database naming - new databases get unique names.
- * fixed node IDs with spaces in (forgot to insert quotes before...)
- * fixed default selection in button link (used to forget current settings)

V1.02beta IMPROVEMENTS

- * save speed now increased by use of buffered I/O
- * added AppWindow icon-dropping for Front Panel window and also document edit window
- * added cursor keycuts in Front panel listview

1.5 Installing Heddley

Installing Heddley could not be easier! Simply double-click on the Install-Heddley icon to install Heddley and documentation to your system. You must have the Commodore-Amiga Installer utility for installation to complete successfully.

If you do not have the Installer utility, then you can download it from Aminet, or find it in various places (such as on some old coverdisks etc).

If you don't want to do that, Heddley can be installed by copying the following files (HEDD: indicates the pathname here where you want to install Heddley - you don't have to make this assign!):

```

Heddley/Archive/Heddley          to HEDD:Heddley
Heddley/Archive/Heddley.info     to HEDD:Heddley.info
Heddley/Archive/Heddley.guide   to HEDD:Heddley.guide
Heddley/Archive/edd.gif         to HEDD:edd.gif
Heddley/Archive/#?.rexx        to REXX:
Heddley/Archive/def_guide.info  to ENVARC:sys/def_guide.info
Heddley/Archive/def_guide.info  to HEDD:Heddley.guide.info

```

(if you don't want to use MagicWB icons substitute def_guide_old.info for def_guide.info and Heddley_old_unsnap.info for Heddley.info).

1.6 Editing the database

When you first start Heddley, you will be presented with the front ↔ panel. If you have AmigaDos 3.0+, [click~here](#) to see a picture of the window.

Function of the window gadgets:

Load

Save

New

Info

Database~documents

Edit

Add

Attrib

Delete

Up

Dn

There is also a menu-strip:

Project

About...

Prefs...

Save~as~HTML

Save~as~ANSI

Save~as~ASCII

Test~Database

Quit

To start editing a database, select either

Load

or

New

. As an example, make a

copy of this document, and then load that into Heddley. Heddley will ONLY display files with a suffix of .guide in its load and save file requesters. Thus you should always name your AmigaGuide® databases with a .guide suffix.

To load a database, in addition to using the

Load

gadget, you can drag and

drop a Workbench icon into the window.

When a database is loaded, or created, the list of documents within that database appears in the Document name listview. These documents correspond to the documents you will see on each page, when read with AmigaGuide® or Multiview. The name of each document MUST BE unique. The reader does not see this name, so it does not have to be that meaningful. The reader will, however, see the document's title (which can be changed with the

Attrib

option).

You can alter each document by selecting it in the listview and then using

Edit

,

Add

,

Attrib

or

Delete

. Double clicking on an item is a shortcut for the

Edit

option. Keyboard shortcuts are available as denoted by the ←
underscored

letter in each gadget. The Esc key will quit Heddley (throughout Heddley pressing Esc has the effect of closing the current window and, where appropriate, selecting the 'Cancel'~action).

When you have done editing your database, be sure to save it again. The text will be justified to the width specified by the

TEXTWIDTH~tooltype

(whose

default is 78 characters).

1.7 Editing a document

The edit window provides the main interface for editing a document ↔
 . The window shows the current text, with buttons highlighted. A scroller is available to move up and down the text (you can also use cursor keys for this too).

If you have AmigaDos 3.0+, [click~here](#) for a picture of this window.

Control is by means of menus. The following menu items are available:

Document	Buttons	Preferences
	Edit~attributes	
	Apply~style~to~text	
	Smooth~scroll	
	Clear~text	
	Make~into~button	
	Load~new~text	
	Delete~button	
	Edit~text	
	Replicate~button	

Exit		

Text can be inserted by means of the options in the Document menu. You can also replace the text in the window with that of your choice by dragging an icon in from the Workbench (obviously it helps if it is the icon of a text file!!!)

You can highlight portions of the text by drag-clicking with the mouse. Once a portion of text is highlighted, you can turn it into a button (using the Make into button menu option), or you can apply a text style (using the Apply style... menu option), such as bold or italic, or change the text coloring. You can highlight a word at a time by double clicking on a word and then dragging the mouse whilst holding the button down on the second click.

To edit an existing button, just click on it, and the
 Button~Edit~window
 will
 opened.

When you have finished editing the document, either press Esc or use the Exit menu option to return to the front panel.

1.8 Other Information

AmigaGuide®

For more information about what AmigaGuide® is, and is about:

1. have a good play with it.
2. read my article on p.108 of the July '94 edition of Amiga User International.
3. obtain the AmigaGuide® developers' distribution (from Aminet, for instance) and read the documentation.

1.9 The Add document gadget

Selecting the Add gadget adds a new document into the database. ↔

Initially the document has a stupid name, which you will do well to change into something more meaningful for yourself. The new document is merely a "blank", but by use of the

Attrib
and
Edit

gadgets you can edit the title and text of this document.

New documents are always added at the end of the document list; the order of the documents is only used for the 'Save as ANSI' feature, where the documents are output in the order they are shown in the listview.

1.10 The Attrib gadget

Selecting the Attrib gadget opens the Document Attribute window. If you have AmigaDos 3.0+ [click~here](#) for a picture of this window.

Function of the window gadgets:

- Doc. ID : this option controls the name of the document as seen in the listview gadget of the front panel. Note that this **MUST BE UNIQUE** for AmigaGuide® to recognise your document. You cannot change the name of the MAIN document as every database needs a MAIN document as a starting point.
- Title : This gadget shows the title which will appear in the title bar when you are viewing this document. Alter this so it reflects the contents of your document.
- TOC : Every document can have an individual contents document, if it is desired. The default contents page is MAIN but you can use this gadget to change it if you want. Note that a contents page is specific to a particular document.
- Next : When using the "Browse" buttons in AmigaGuide, you can

specify which document follows the current one. Select the name of the document to follow, or «none» for the default successor (as in the listview).

Prev : When using the "Browse" buttons in AmigaGuide, you can specify which document is moved to using the "Browse <" button. Select the name of the document to follow, or «none» for the default predecessor (as in the listview).

Use Esc, 'Cancel' or the close gadget to close this window and cancel any changes you have made, or 'Okay'~to keep them.

1.11 The Delete gadget

Selecting this gadget will remove FOR EVER the current document (as displayed at the base of the listview) from the database. USE WITH CARE!

1.12 Clear text

Selecting this option will wipe all the text and buttons in the current document, leaving you with a blank document.

1.13 Load text

Selecting Load text will enable you to load a new text file into the document window. All current text and buttons will be cleared. Buttons can then be added into the text to produce your document.

1.14 Edit text

This option allows you to edit the actual text of the document.

Heddley compiles the current document into AmigaGuide@ format, and invokes your text editor with the compiled text as its argument. The default editor is "ed", but you can change this by setting the EDITOR environmental variable, or altering the

```
EDITOR~tooltype  
, before you start up Heddley. Heddley starts the  
text editor by performing the command:
```

```
<editorname> <temporaryfile>
```

The contents of the current document are placed in the temporary file, which you must save back before you quit the editor. Do NOT change the name of this file.

Also, your text editor must run SYNCHRONOUSLY. This is because Heddley waits

until the editor has finished and then reloads the document. If your editor spawns another process then Heddley will immediately reload the document and not wait; obviously not very desirable. If you use an asynchronous text editor, you need to write some

```
ARexx~scripts
to coordinate your editor with
```

Heddley.

You will notice that buttons are translated into their AmigaGuide® form in this text, this is in order to preserve them during the edit. Unless you are sure of yourself, don't touch these definitions (these are the bits of text with a @{ at the beginning and a } at the end).

When you finish editing, select the save-and-quit option from your text editor. Heddley will then redisplay your edited text. Note that this may take a small while for big files; Heddley's retranslating mechanism is a little slow at the moment, but it enables Heddley to do 'the right thing'~by spaces.

1.15 Make into button

```

Selecting this option allows you to create a button in your ↔
document. To use
```

it you must have some text highlighted (use the mouse to do this). Heddley will then create a button and open the

```
Button~Edit~window
```

.

Normally, AmigaGuide does NOT allow multi-line buttons. However, if you select more than one line in Heddley and turn it into a button, Heddley will translate this into a button for each line. When you reload the database (or edit the document), each line will have its separate button. But for the first time (ie. just after you created the multi-line button) all the line-buttons will be amalgamated into one button. There should be no visible difference!

1.16 Delete button

This option allows you to remove a button. Simply select this option, and then click on the button you wish to remove. If you decide not to remove a button after all, reselect the Delete button menu-option to cancel the operation.

1.17 The Button Edit window

```

The Button Edit window allows you to alter the nature of the ↔
button you are
```

editing. If you have AmigaDos 3.0+, click~here to see a picture of the window.

Function of the window gadgets:

```
Button type - the button type determines what will happen when the
```

- reader clicks on this button. See Button Types for more information.
- Command - this is the name of the AmigaDOS/ARexx command (or ARexx script) that will be executed when the reader clicks on this button. Only applicable if button type is one of AmigaDOS command, ARexx command, or ARexx script.
- Document - this is the name of the document which the current button will take the reader to, when clicked. If it is an internal link (see Button Types) then it will be the name of another document in the current database.
If it is an external link then it can be any text file or AmigaGuide® file (WB2) or any Workbench 3 datatype supported file.
- Line # - this is the top line of the document that will be displayed with a document/text file link.
- New wnd - if this gadget is checked, then the link will cause the linked document to load in a NEW WINDOW of AmigaGuide. Try this and see!
- Choose Doc - Click on the Choose Doc button to bring a list of documents (internal link) or a file requester (external link) to choose the link document from.
- Create Doc - Click on this button to create a new document there and then. This is useful as you often think of the need for a new document whilst writing an existing one. Clicking here will add a new document and open the

Attrib~Window
for the newly created document.

Use Esc, 'Cancel'~or the close gadget to cancel your alterations, or 'Okay' to keep them.

1.18 Load database

Use of this gadget will open a file requester to load a new database. The database must have a .guide suffix. ←

In the case of malformed AmigaGuide databases (ie. those with various syntax errors and so on) Heddley will try and do its best, although there is a possibility that very malformed files will crash Heddley.

If the database has no name then it will be given the name "Unnamed_Database". Documents with no name get given one, documents with no title get a copy of the name as a title. If the database you load does not have a "Main" node, Heddley will add one for you. Links with no destination become links to "Heddley_Error".

However, Heddley can be temperamental and if you do find a .guide which causes Heddley to either crash or make Enforcer hits, please please please mail me with a copy of the .guide that caused the problem and a description of the problem.

1.19 Save

This gadget will open a file requester in order for you to choose the file to save the database to. Be sure to give your file a .guide suffix.

1.20 The Info Window

The Info window allows you set various database-wide attributes: Index, Help document, Author, Version, Copyright notice, wordwrap and database font. If you have AmigaDos 3.0+, click~here to see a picture of this window.

Index

The Index gadget will allow you to set the index document for the current database. The Index node will be displayed when the reader clicks the Index gadget in AmigaGuide® or Multiview.

Select « none » to disable the index (this is the default).

Select « auto » to make Heddley create the index automatically for you. For an example of this, see the

index to this database (press the Index gadget or select Index from the Contents page). Heddley creates an alphabetical list of document titles, followed by an alphabetical list of all links within document (this includes ARexx commands etc., although I may change this later if I change my mind...)

The index document is given the name HeddleyAutoIndex, so avoid using this name for your own documents, and if you want to create extra links to the index, enter HeddleyAutoIndex in the internal link

document~name~gadget
for a

button.

Help

The Help gadget will allow you to set the help document that is displayed when the user presses HELP in AmigaGuide/MultiView or selects the Help gadget. Either select a document in your database, or « default » to use the normal~AmigaGuide~help~file, which is normally stored in "s:help.guide".

Author

The Author gadget allows you to edit the recorded author of the database. For a new database, the default value of this field is the contents of the AUTHOR environmental variable.

(C)

The (C) gadget allows you to edit the copyright message of the database. For a new database, the default value of this is the contents of the COMPANY environmental variable.

Version

The Version gadget allows you to keep version control on your database. On creation of a new database, this field is set to '1.0'. The version of a database created this way can be found by using the C:Version command:

```
version mydatabase.guide ~~~~~
```

Wrap

The Wrap gadget allows you to use the

```
AmigaGuide~V39
wordwrap facility. This
```

lets AmigaGuide do the formatting, instead of Heddley. When this option is used, you can also choose the font for your database with the Font gadget. I advise against using this in documents which are meant to be read in anything but AmigaGuide V39+ as it makes documents virtually unreadable for Workbench 2.x users.

Of course, if you are a Workbench 2.x user and come across someone else's database written this way, you can easily use Heddley to convert it to a more readable form! Ain't life sweet?

1.21 Starting a new database

Selecting this gadget will wipe all the current database, and start a new database. The MAIN node will be automatically created for you.

1.22 The document name list

This list shows the unique IDs of each document in the database. ↔

Use the

scroller to move up and down this list. Double-clicking on an entry will open the

```
edit~window
for that document.
```

1.23 Starting Heddley

Heddley can be run from either the Workbench, or the command line. ↵
From

Workbench:

Double-click on Heddley's icon. If you have shift-clicked a guide file beforehand, that file will be loaded into Heddley on startup.

From the Shell:

Change to the directory where Heddley is stored, and type "Heddley". Heddley accepts command line arguments, similar to the tooltypes available in the icon, but parsed according to the following template:

```
FILE,
    FONT
    /K,
    FONTSIZE
    /K/N,
    TEXTWIDTH
    /K/N,
    EDITOR
    /K,

    FGPEN
    /K/N,
    BGPEN
    /K/N,
    HELPFILE
    /K
```

For example:

```
heddley FONT Courier.font FONTSIZE 13 EDITOR "vi" myguide.guide
```

...will start up Heddley using 13pt Courier, "vi" as the default editor and automatically load in "myguide.guide" for editing.

1.24 Configuring Heddley

Heddley can be configured in three ways:

- * the 'Prefs...' option from the main menu
- * the ARexx~port
 - * the tooltypes of Heddley's icon

Heddley processes the configuration in the following order:

```
'Prefs...'  
settings _override_ tooltypes
```

Heddley's configuration can be stored in the tooltypes of Heddley's icon. To view these, select Heddley's icon and select 'Information' from the 'Icons' menu of Workbench. Here is a list of the tooltypes:

```
FONT  
  
FONTSIZE  
  
TEXTWIDTH  
  
EDITOR  
  
FGPEN  
  
BGPEN  
  
HELPPFILE  
If you start Heddley from the  
Shell  
, these tooltypes can be used as parameters
```

for Heddley.

The values of these tooltypes can be found using the
ARexx

```
QUERY  
command when
```

Heddley is running.

1.25 The ARexx port

Heddley's ARexx port offers all the functionality of the front GUI \leftrightarrow panel, and more besides. The portname is HEDDLEY.N, where N is the instance of Heddley you wish to control. This will usually be 1.

Here is a list of all the commands that Heddley supports:

```
~ADD~~~~~  
  
~ATTRIB~~  
  
~DELETE~~  
  
~EDIT~~~~  
  
~GOTO~~~~
```

```
~INFO~~~~
    * changed from INDEX since v1.0

~LOAD~~~~

~LOCK~~~~

~MOVE~~~~

~NEW~~~~

~PENS~~~~

~PREFS~~~
    * new for v1.1

~QUERY~~~
    * new variables for v1.1

~QUIT~~~~

~SAVE~~~~

~TEST~~~~
    * new for v1.1
```

Using the ARexx port will allow you to interface Heddley with an asynchronous text editor; see the examples for more information.

The on-line help facility also offers an AmigaGuide~AREXX~port
.

1.26 The FONT tooltype

This tooltype selects the font for the document~edit~window
. It must be a fixed-width font. Default:

```
FONT=topaz.font
```

Don't forget the .font suffix!

If your font cannot be found, or if it is proportionally spaced, then the default will be used.

1.27 The FONTSIZE tooltype

FONTSIZE

This tooltype selects the font size for the document edit window. Default:

```
FONTSIZE=8
```

If the needed size does not exist, then the nearest size available will be scaled up/down as needed.

1.28 The EDITOR tooltype

```
This tooltype selects the text editor to be used when
editing~a~document's
```

```
contents
. Default:
```

```
EDITOR=ed
```

The text editor must run synchronously (not spawn its own process). If you use GoldEd, the recommended setting is:

```
EDITOR=ed sticky
```

Where the 'ed' is the GoldEd quick-start editor.

If the EDITOR tooltype is NOT specified, then if the EDITOR environmental variable is set, it this will be used instead.

1.29 The TEXTWIDTH tooltype

This tooltype selects the normal width of text justification, both for the edit window and outputted documents. The default is:

```
TEXTWIDTH=78
```

If the font or text width you choose makes the text too wide to display in the window, the largest possible width will be used for the window display.

TEXTWIDTH has a minimum value of 1; and using a value of 1 is not a good idea as there may be some bugs in Heddley's output routines at the moment which lead to strange output source. You have been warned!

It is not recommended that you change TEXTWIDTH from 78, as this is the most number of columns that can be displayed on a normal screen with topaz-8 as the font.

It is ALWAYS better to justify to 78 characters, for the benefit of Workbench 2 users. Be nice!

1.30 The HELPFILE tooltype

This tooltype enables you to specify where Heddley should look for ↵ its documentation file for the on-line~help facility. For example, if you like to keep all your documentation in a partition called 'MAN:' you might specify:

```
HELPFILE=MAN:Heddley.guide
```

The default is:

```
HELPFILE=Heddley.guide
```

This means Heddley will look for the file Heddley.guide in the directory that it was started from.

1.31 The FGPEN tooltype

FGPEN controls the logical number of the pen used to render the text in the document edit window. The default is:

```
FGPEN=1~
```

1.32 The BGPEN tooltype

BGPEN controls the logical number of the pen used to render the background in the document edit window. The default is:

```
BGPEN=0~
```

1.33 ADD

COMMAND

```
ADD
```

SYNTAX

```
ADD NAME/K,NEW/S
```

DESCRIPTION

Adds a new document into the document list.
If a NAME is specified, that name is used for the new document,
otherwise if NEW is specified, a unique name is automatically
generated. ~

1.34 ATTRIB

COMMAND

ATTRIB

SYNTAX

ATTRIB NAME/K, TITLE/K, TOC/K, GUI/S, NEXT/K, PREV/K

DESCRIPTION

ATTRIB changes the attributes of the current document. If NAME, TITLE TOC, NEXT or PREV are specified then the relevant attributes are updated:

NAME	ASCII unique document name
TITLE	ASCII title of document
TOC	ASCII document name for document table of contents
NEXT	ASCII document name of Browse > successor
PREV	ASCII document name of Browse < predecessor

If GUI is specified then Heddley will open the
 Edit~Attribute
 window
 and let the user change the attributes.~~~~~

1.35 DELETE

COMMAND

DELETE

SYNTAX

DELETE NAME/K, NUMBER/K/N, CURRENT/S

DESCRIPTION

DELETE deletes a document from the document list.

CURRENT deletes the current document.

NAME deletes the document with specified name.

NUMBER deletes the document specified by position in the list.

1.36 EDIT

COMMAND

EDIT

SYNTAX

EDIT LOADTEXT/K,PLAIN/S,GUI/S,SAVETEXT/K

DESCRIPTION

EDIT allows editing of the current document text. If GUI is specified, the

document~edit~window
is opened and an interactive editing session
started.

LOADTEXT and SAVETEXT allow the contents of the document to be loaded from and saved to a file. In LOADTEXT, the file contents will be interpreted as an AmigaGuide® file, and in SAVETEXT AmigaGuide® control information will be saved in the text. This feature is useful for interfacing Heddley with asynchronous editors. If you just want the plain text to be loaded/saved and buttons ignored, specify the PLAIN switch.~

1.37 GOTO

COMMAND

GOTO

SYNTAX

GOTO NAME/K,NUMBER/N/K

DESCRIPTION

GOTO selects a document to be the current document in Heddley. If NAME is specified then the node of that name is activated. Note that the search is case insensitive. If NUMBER is specified, then the document with the specified position in the listview is made the current one.

1.38 INFO

COMMAND

INFO

SYNTAX

INFO INDEX/K,HELP/K,AUTHOR/K,COPYRIGHT/K,VERSION/K,NOHELP/S,
AUTOINDEX/S,NOINDEX/S,GUI/S

DESCRIPTION

The INFO command is used to alter global attributes of the database.

Indexing:

INDEX sets the index document name.
NOINDEX specifies no index is to be used for this database.
AUTOINDEX enables automatic~index~creation for this database.

Help:

HELP sets the help document name.
NOHELP specifies AmigaGuide default help to be used.

Version control:

AUTHOR sets the database's author.
COPYRIGHT sets the copyright notice of the database.
VERSION sets the version string of the database.

Other:

GUI opens the info~window to allow the user to choose and see the attributes.

NOTE

This command supersedes the INDEX command from version 1.0 of Heddley.

1.39 LOAD

COMMAND

LOAD

SYNTAX

LOAD FILE,CONFIRM/S,REQUEST/S

DESCRIPTION

LOAD loads a new database into Heddley's memory. CONFIRM asks the user if they're sure they want to load a new database, if they have made changes to the current database. REQUEST puts up a file requester on the screen to get the file to load from the user.

1.40 LOCK

COMMAND

LOCK

SYNTAX

LOCK UNLOCK/S

DESCRIPTION

LOCK locks the front panel of Heddley so the user cannot use any of the gadgets. It also changes the pointer to a wait-pointer. Using the UNLOCK argument unlocks the window.

(Un)locking a window more than once has no ill-effects.

BUGS

At the moment, LOCK does **not** prohibit AppWindow behaviour, so it is still possible to load a database using icon-dropping while the front panel is locked.

1.41 NEW

COMMAND

NEW

SYNTAX

NEW CONFIRM/S

DESCRIPTION

NEW wipes the current database and initialises Heddley with a new database. A new database consists of an empty Main document. Specifying CONFIRM option will ask the user if they are sure they want to wipe the current database if it has been modified.

1.42 PENS

COMMAND

PENS

SYNTAX

PENS FGPEN/K/N,BGPEN/K/N

DESCRIPTION

PENS allows the setting of the pens used to render the button edit window. The functions of the parameters

FGPEN

and
 BGPEN
 are
 as for their corresponding tooltypes.

1.43 QUERY

COMMAND

QUERY

SYNTAX

QUERY ATTRIB/A,VAR/K/A

DESCRIPTION

QUERY retrieves the value of an internal Heddley variable (attribute) and places it in the ARExx variable specified after the VAR keyword.

Example:

```
QUERY FILENAME VAR THEFILE
SAY 'Current filename is:' THEFILE
```

HEDDLEY ATTRIBUTES

AUTHOR	Author of the current database. Set using INFO AUTHOR.
BGPEN	Pen used for background in button edit window. See BGPEN~tooltype . Set using the PENS command.
BUTTPEN	Pen used for button highlight in button edit window. See BUTTPEN~tooltype. Set using the PENS command.
CHANGED	TRUE if database has been changed since last load/save operation, FALSE otherwise.
COPYRIGHT	The copyright notice of the current database. Set using INFO COPYRIGHT.
CURRENT	The name of the current document (as shown in the listview in the front panel) or empty if there is none. See GOTO

and
ATTRIB
.

DATABASE The name of the current database.

DBVERSION The version of the current database. Set using
INFO
 VERSION.

DIRNAME The name of the current directory of the current
database.

DOCFILENAME The temporary filename of the current document text.
You should not normally need this.

DOCNEXT The name of the browse-successor document, if set.
Set with
ATTRIB
. See also the
Attrib~window
.

DOCPREV The name of the browse-predecessor document if set.
Set with
ATTRIB
. See also the
Attrib~window
.

DOCTITLE The title of the current document. Set with
ATTRIB
.

DOCTOC The table of contents of the current document, or empty
if default toc. See
ATTRIB
.

EDITOR The text editor used for document editing. See the

EDITOR~tooltype
and
Edit~Text
in the button edit
window.

FGPEN The pen used for rendering text in the document edit
window. See
FGPEN~tooltype
. Set using the
PENS
command.

FILENAME The filename (relative to the current DIRNAME) of the
current database. Set according to the most recent load
or save operation.

FONTNAME The name (.font extension compulsory) of the font used
 in the button edit window. See the
FONT~tooltype
.

FONTSIZE The Y-size of the font in the button edit window.
 See the
FONTSIZE~tooltype
.

HELP The name of the help document in the current database.
 Set using the
INFO
 HELP command.

INDEX The name of the index document in the current database.
 Set using the
INFO
 INDEX command.

INFORMATION Full version string.

ISMAIN TRUE if the current document is the Main document.

LOADED TRUE if a database is being edited, FALSE if not.

MASTER The *full* pathname of the database being edited (note
 that relative pathnames are allowed; relative to the
 directory Heddley was started from).

TEXTWIDTH The width of the justification in the button edit
 window.
 Set using the
TEXTWIDTH~tooltype
.

VERSION The version number of Heddley. Use this if you want to
 write ARexx scripts which need a new version of
 Heddley.

1.44 QUIT

COMMAND

QUIT

SYNTAX

QUIT CONFIRM/S

DESCRIPTION

QUIT causes Heddley to quit. If CONFIRM is specified, then the user is asked to confirm the quit action if they have not saved alterations

to the current database.

1.45 SAVE

COMMAND

SAVE

SYNTAX

SAVE FILE,REQUEST/S,NOICON/S,HTML/S,ANSI/S,STRIP/S

DESCRIPTION

SAVE causes Heddley to save the current database. If the FILE argument is specified then that argument overrides the current filename.

If REQUEST is specified then a file requester is opened and a filename obtained from the user.

If NOICON is specified the creation of an icon for the current database is suppressed. Heddley attempts to use

ENVARC:sys/def_guide.info

as the database icon, or else the default project icon. Note that if an icon already exists, it will never be overwritten, irrespective of whether NOICON was specified.

If HTML is specified, then the database will be saved out as HTML, not AmigaGuide, source. See

Saving~HTML~source

.
As Heddley saves directories for HTML source, the icon saved will be the default drawer icon.

If ANSI is specified then the database will be saved out as ASCII with ANSI standard control sequences for colors and text style. See

Saving~ANSI~source

. The icon saved will be the default project icon.

If ANSI is in use and STRIP is specified, then the ANSI color codes will be removed from the output, and just plain ASCII saved.

1.46 TEST

COMMAND

TEST

SYNTAX

TEST

DESCRIPTION

TEST causes Heddley to output the current database to a temporary file (of the format 'T:<unique-number>.tguide'). It then tells AmigaGuide to display that database.

Database viewing is an asynchronous operation.

See the menu item,
 'Test~Database'
 .

1.47 PREFS

COMMAND

PREFS

SYNTAX

PREFS LOAD/S,SAVE/S,GUI/S

DESCRIPTION

PREFS LOAD loads the preferences from S:Heddley.prefs
PREFS SAVE saves the preferences into S:Heddley.prefs
PREFS GUI opens the
 preferences~window
 and obtains preferences
settings from the user.

1.48 MOVE

COMMAND

MOVE

SYNTAX

MOVE UP/S,DOWN/S

DESCRIPTION

MOVE moves the current document one place up or down the document list, according to the specified arguments.

1.49 ARexx examples

Example ARexx scripts

Some ARexx scripts to work with Heddley and other applications have been written. To view them, click on the relevant button below. (If the scripts do not live in REXX: type ASSIGN REXX: <rexmdir> ADD at the CLI in order for the below to work - where <rexmdir> is where the Heddley ARexx scripts are stored).

editnode.rexx

- Grabbing a document from Heddley into GoldED

savenode.rexx

- Writing a document back from GoldED to Heddley

databaseinfo.rexx

- Prints information about current Heddley database
CLICK~HERE to run this script (only works if you've got
the scripts in REXX:)

hedd_empty.rexx

- An empty template for writing Heddley ARexx scripts in

hedd_golded_temp.rexx

- An empty template for writing GoldED - Heddley ARexx scripts in

1.50 Distribution of Heddley

Heddley is (C) Edmund Dumbill 1994.

Heddley is SHAREWARE. It is NOT public domain. The author gives permission for Heddley to be freely distributed providing that no commercial gain is made from its distribution, and no modification is made to the original distribution.

Organisations or people wishing to include Heddley on a coverdisk or CD-ROM should contact me first to sort out the details. This EXCLUDES the AMINET CD-ROM or Fred Fish CD-ROM collections.

DISCLAIMER

This software is provided "as-is" and the
author
accepts no responsibility for
damage and/or loss of data/equipment resulting from the use of this software.
(not that it's very likely! ;-))

COST

No features of Heddley are crippled to allow you to appreciate the program to the full. If you keep Heddley for more than an evaluation period, please send the shareware fee to the author. Please send international money orders or, if you're willing to risk it, cash. Cheques are only acceptable from within the UK. Please make cheques/orders payable to "Edmund Dumbill".

Shareware fee:

UK pounds	10.00
US Dollars	15.00
DM	25.00

(or equivalent in currency)

Contact address:

Edd Dumbill,
13 Giles Avenue,
Burnholme,
York. YO3 0RB.
England.

(valid until end of June 1995).

Please pay the fee: Heddley has taken a LOT of my valuable time. Consider how much time you are saving yourself by using it! Paying the fee will really encourage me to continue developing Heddley.

As an alternative to paying the fee, if you've written any *decent* Shareware yourself send me a registered copy (by snail-mail, or UUencode an LHA archive and email it to

me
).

1.51 Saving HTML source

Heddley can save your AmigaGuide database as HTML source, for use with Mosaic and other WWW browsers, by use of the 'Save as HTML' menuitem from the front panel.

In this case, the filename you specify will be a DIRECTORY. HTML needs a directory for each database. The main document will be saved as Main.html, and the rest of the documents as <doc_id>.html where <doc_id> is the unique document ID, as in the

main~listview

. To test out your generated HTML file select 'Load local file'~from the Mosaic menu, and load the 'Main.html' file from the HTML directory you saved out with Heddley.

Heddley's layout scheme for HTML is as follows: blank lines in the AmigaGuide document signify end of paragraphs, lines followed immediately by an indented next line are assumed to be preformatted, and the <PRE> directive will be used for the following block of text, and all other lines are wrapped normally. The best thing to do is try it and see.

Differences between AmigaGuide output and HTML output:

HTML does not support contents pages in the same way as AmigaGuide, but in each HTML document an anchor has been provided on the first line which will take you to the table of contents. Likewise for

indexes.

HTML does not (afaik) support underline as a text style. Where underline is used in AmigaGuide databases, it will be replaced with the typewriter style in HTML. You may not like this behaviour, so tell

me
if you don't! Bold and italic are supported fully.

HTML eats leading spaces, so indenting is not possible in the same way as is in AmigaGuide, say on this page, and as lists are not supported yet, indenting is not yet possible in HTML.

HTML cannot handle AmigaDOS commands or ARexx commands; they don't make much sense as HTML is not machine-dependent!

Links to external files will be interpreted as follows:

link fullpathname/main will be translated to
href=file://localhost/fullpathname

Note that this means your external links must be ABSOLUTE pathnames.

HTML features not supported yet:
Inlined images, lists, etc, etc.

Comments and suggestions will encourage me to keep improving HTML support though. If enough people ask for it I'm quite amenable to improving the HTML facilities. Mail

me
and see!

1.52 The Preferences Window

Heddley can be configured by use of the prefs window. If you have ↔
AmigaDos
3.0+, click~here to see a picture of this window.

Function of the window gadgets:

Font

Font~size

Palette

Current~pen

Editor~string

Text~width

Use the 'Save' gadget to save the configuration to Heddley's prefs ↔
file

(S:Heddley.prefs). The current position of all the windows will also be saved,

so you can move the windows to positions that suit your monitor and screen setup, and these positions will be used by Heddley from then on.

Use the 'Use' gadgets to keep the preferences for this run of Heddley only, and the 'Quit' gadget to cancel any changes to the preferences you have made.

1.53 The Palette Gadget

This gadget allows you to set the color of the
current~pen
.~

1.54 The Current Pen cycle gadget

This cycle gadget controls which pen color is currently being ←
edited.
Currently there are two pens in use, the Text Pen which corresponds to the

FGPEN
tooltype and the Back Pen which corresponds to the
BGPEN
tooltype.

1.55 The Font Button

Clicking on the Font button will enable you to choose a new font ←
for the

document~edit~window
from a font requester. The font name and font size will
be updated when you have made your choice.

This option corresponds to the
FONT
and
FONTSIZE
tooltypes, and similar
restrictions apply.~

1.56 Document Pick List

This selector is used by different functions to obtain the name of a document. Simply double click on the document you wish to select, or close the window to keep your previous choice.~~

If you have AmigaDos 3.0+, click~here

1.57 Test Database

Selecting Test Database will output the current database you are editing to a temporary file, and then tell AmigaGuide to load that database. As for the

on-line~help function, you can view the database at the same time as using all the functions in Heddley.

The temporary file is removed when you quit Heddley.

1.58 On line help

Heddley provides on-line help through AmigaGuide. To obtain help in any part of the program, simply press the HELP key. Heddley will then open an AmigaGuide window and cause the relevant part of the documentation to be displayed.

Viewing the help file is performed asynchronously from Heddley; that is you can still use Heddley while the help window is open.

The tooltype/command line argument

HELPPFILE

tells Heddley where to find the help database; if neither are specified, the default is 'Heddley.guide' in the same directory as Heddley was started from. If Heddley cannot find the help file, the message "Can't open database" will be displayed and you will not be able to access on-line help.

Note that to use this function you need amigaguide.library V34+ and the help file Heddley.guide.~

1.59 AmigaGuide ARexx Port

If amigaguide.library V34+ is available, Heddley will have an ARexx port for the AmigaGuide interface too. It will be named as HEDDLEY_HELP.N, where N is the instance of Heddley (starting with 1). The commands available at this ARexx port are:

- | | | | |
|--------|---------------|---|--|
| ALINK | DOCUMENT,LINE | - | load the named document into a new window, at line LINE. |
| CLOSE | | - | close the window (only use on windows opened with ALINK). |
| LINK | DOCUMENT,LINE | - | load the named document into the current window, at line LINE. |
| SYSTEM | COMMAND | - | execute a system command. |

QUIT - shut down the current database.

1.60 Quick start guide to using Heddley

This section will give you a quick (but not very comprehensive) guide to creating your first AmigaGuide document using Heddley. ←

Step 1 - Create a database

After loading Heddley, click on the 'New' gadget. This will create a new database. Then click on the 'Attrib'~gadget; you are now editing the attributes of the first document. Enter a suitable title in the title gadget, e.g. 'My First Database'. Close the
Attrib~window
.

Step 2 - Putting text in the document

Select 'Edit'~and the
document~edit~window
will open. You can now either edit this text or load some in fresh. If you haven't set up the interface to your text editor properly yet, use the 'Load new text'~menu option and load in a text file (any one will do!). Otherwise you can use the 'Edit text' menu option and type your own text in.

Step 3 - Creating buttons

It's time to create a button in your document. Select with the mouse a word in the window. Then select the 'Make into Button' menu option. The
button~edit
window
will now open. We want to create a document for this button to ←
link to,
so click on the 'Create' button. The
Attrib~window
will open for this new
document. Alter the Doc. ID attribute to something meaningful, and alter the title if you want. Close the Attrib window. Close the button edit window. Hey presto! You've just created your first link.

Step 4 - More text...

Close the document edit window. Select the document you just created (the one that the button links to). Select 'Edit'~and, as in step 2, put some text into this document. Close the document edit window.

Step 5 - Saving

Select 'Save'~from the front panel, and choose a filename and destination, say 'RAM:myguide.guide'. Now quit Heddley, open the RAM: disk from Workbench, and double-click on the created file. (If this doesn't work then your def_guide icon needs setting up properly). MultiView should load your new database in.

OK, it's not very impressive, but you should get the idea of how to create databases.

Step 6 - Challenge

If anyone feels they can write a better tutorial than this (it can't be hard!) then if they mail it to

me
I'll include it in the distribution!

1.61 Things I plan to implement in Heddley

To do

Obviously, Heddley is not complete. There is much room for expansion in Heddley. Features planned for the future are:

Definite

- * implement @keywords

Probable

- * add margin-adding and stripping to format text nicely for lower end systems. Also variable document indent.
- * add grouping of documents to make writing bigger databases much easier. [ghh]
- * add more HTML markup control - maybe an 'object'~placement system where objects can be lists, images, program listings etc.
- * add 'save-as-TeXInfo' feature
- * support locale.library [bagder]
- * rationalise ctrl-C handling [bagder]

If you have any reasonable suggestions, please mail them to the author

.

If you have an MMU and find any Enforcer hits, the

author

would be **very**

pleased to receive full details of them!~~~~~

Bugs

There are bound to be bugs in a program this size, and I'd be grateful if you would send detailed reports for any bugs you do find. As ever, be careful about backing up your data!

1.62 Moving a document - Up

Clicking on the "Up" gadget will move the current document one place up the listview. Use SHIFT-Cursor-Up as a shortcut for this.

1.63 Moving a document - Down

Clicking on the "Dn" gadget will move the current document one place down the listview. Use SHIFT-Cursor-Down as a shortcut for this.

1.64 Change text styles

The Style window allows you to alter the style of the text you have currently highlighted. When the window opens, it will show the current state of the character at the beginning of the highlighted portion of text. If you use the Cancel button then no change will be effected. Note that only AmigaGuide~V39 (Workbench 3.x+) can handle text control sequences.

If you have AmigaDos 3.0+, then click~here to see a picture of the window.

Explanation of the window's buttons:

Foregnd text: this button controls the pen used for drawing the text, it can be one of Text, Shine, Shadow, Fill, Filltext, Background and Highlight.

Backgnd text: this buttons controls the pen used for drawing the background beneath the text. It has the same choice of pens as the foreground.

Bold: this checkbox determines whether the bold style will be applied to the highlighted text.

Italic: this checkbox determines whether the italic style will be applied to the highlighted text.

Underline: this checkbox determines whether the underlined style will be applied to the highlighted text.

Reset: this button resets the colors and styles to what they were when the window was opened.

Cancel: this buttons aborts the operation - no style will be applied to the text if this is pressed.

1.65 What About...?

This section deals with any popular queries about Heddley. Sometimes the way Heddley behaves is a little peculiar, and explanations for this will appear here. If you want to see an answer here, mail me and I'll write it...

Formatting of text

Heddley reformats any text that is loaded into it. The reason for this is to

support AmigaGuide V34 where the @wordwrap option is not available. The following rules are followed by Heddley:

- * if a line ends in a line feed, then:
 - if the next character on the first line is a space or tab then the line feed is obeyed
 - if the next line is a line feed, then both line feeds are obeyed
 - if the next line is any other character (text) then the line feed is ignored and the lines are concatenated and justified

If you have a table, say, on which every entry must start on a new line, make sure that the table is indented at least one space (4 spaces look nice) from the left hand side. Heddley will then respect the formatting (as it did for the bulleted list above).

Tabs are currently blindly expanded to 8 spaces. The answer is - don't use tabs, they're a pain!~~~

Fonts

Heddley leans heavily towards writing AmigaGuide files using fixed width fonts. The reason for this is that I think it's important to maintain compatibility with the lowest current OS in widespread use, ie. OS2.04. 2.04 AmigaGuide only supports fixed width fonts, so we only edit in fixed width fonts. However in the next version of Heddley, support for proportional fonts may well be included in parallel with the @wordwrap option. As far as I can see, the main document edit screen will always use fixed width fonts though.

1.66 AmigaGuide V39+

AmigaGuide v39 (Workbench 3.x+ versions) supports more commands than v34. In particular Workbench 2 users will not be able to use text control sequences like bold, italic etc. These sequences should appear on WB2 as faint white lines, and should not bother WB2 users too much.

Heddley does not yet support all commands, at the moment the following are supported:

```
@{b} @{ub} @{u} @{uu} @{i} @{ui}
@{fg <pen>} @{bg <pen>} @wordwrap @font
@author @(c) @$VER:~~
```

1.67 Why should I use Heddley?

Here is a list of the main features of Heddley, to let you see at a glance what it can do for you.

Feature list (+ = new for version 1.1)

- + Automatic index creation option

- + 'Test-database' facility
 - + On-line context sensitive AmigaGuide help by pressing HELP key
 - + Export as HTML feature
 - + Export as ASCII/ANSI text feature
 - + Full text control - bold, italic, underline & colors - loading, editing and saving
 - + Preferences Edit window
 - + Supports new version control commands @author, @\$VER: and @(c)
 - + Supports V39 layout commands @font, @wordwrap
- Full featured ARexx port
- Neat and good looking user interface
- Fully font adaptive & size adaptive (to a measure ;)
- AppWindow icon-drop support
- Interfaces with your favorite text editor
- Easy to install
- Distribution includes detailed documentation and example ARexx scripts
- Works with Workbench 2 or 3

1.68 How do I?

This page provides information on how to do useful things with ←
Heddley.

How do I...

...use my own index?

To use your own index document, open the
Info~window
from the front panel, and
alter the 'Index' attribute to read the name of your document.

...add my own help?

To use your own help document open the
Info~window
from the front panel, and
alter the 'Help' attribute to read the name of your document.

...use

```
AmigaGuide~v39
wordwrapping?
```

If you must use this feature (it's very unfriendly to Workbench 2.x owners), open the

```
Info~window
from the front panel and adjust the 'Wrap' checkbox
```

accordingly.

...choose a font for my database?

If you want AmigaGuide (v39+) to use a particular font for your database, open the

```
Info~window
, ensure the 'Wrap' box is checkmarked, and then select the
'Font'~button to choose the font. It's best to choose a font you can be sure
other people will have, such as those provided as standard with Workbench
(Times, Helvetica etc). Please don't use this option if you want Workbench 2.x
owners to be able to read your databases. ~~
```

1.69 Replicate button

Selecting the menuitem 'Replicate button'~has the effect of replicating the next button you click on. Select 'Replicate button' again to cancel the operation.

Replicating a button has the effect of scanning the text following a button for identical text as the selected button, and creating a button there too. For example, if you have a document with more than one instance of the word "Amiga" which you want to link to another document, create a button from the first instance of "Amiga", then select the 'Replicate button' menuitem, and then select the button you just created.

1.70 Smooth scroll

The menuitem 'Smooth scroll' allows you to toggle on and off the use of smooth scrolling in the document edit window. Useless really :-)

1.71 Save as ANSI

Heddley offers the feature of saving your database as a text file, ←
with ANSI

control sequences. Most text viewers will be able to interpret these (in particular the excellent "Most" by Uwe Röhm), as well as your printer driver. Select the 'Save as ANSI' menuitem to save the database in this way.

The documents will be output to the text file in the order they are shown in the front panel listview, so use the


```
'Up'
and
'Dn'
gadgets to change the order
```

to a meaningful one.

The following layout is used: Heddley will generate a title page from the information as given in the

```
Info~window
```

```
. Each page will be numbered in the
```

center of the bottom of the page. Each page will have a running header, which will be the title of the current document. Both headers and footers will be in ANSI 'faint'~style. Every page will be terminated with a form-feed. When a new document begins, it is given a document number and its title will be output bold in color number 3. If the

```
auto-index
```

```
option is being used, the index will
```

be output at the end of the file. The index has two parts; an index of sections (with the references being of the form P.N where P is the page number and N the line number) and an index of keywords (what were buttons) with references of the same form.

Heddley will attach your default project icon (ENVARC:def_project.info) to the file that is created.

1.72 Writing Amigaguide files

Here are some thoughts I have on how to write your AmigaGuide ←
databases with

Heddley. They're just my opinions, but they may help you in writing your own databases. They apply more perhaps to program documentation than other sorts of databases, but they're still worth reading. I am aware too, that I have not followed my own guidelines all the way through this documentation. This is because some of them have come out of the experience of writing this file, and also this database has evolved over a considerable period of time.

The Main document and subdivision of information

Ideally, the Main document should be a contents page. Resist the temptation to make it a title page; this will just annoy users who use the documentation regularly. Subdivide your information into chapter-sized portions; as if you were writing a book and you needed to divide the information up. It's important that all information that could be grouped is available off the same link in the contents page; there ought to be a logical route to the information that the user wants without them having to search for it. Make sure that your Main document is not too sparse; having good looking pages is just as important as a good looking program, as using AmigaGuide gives your documentation a "user-interface". The reader would still like to know the title and description of the database, and added information about each "chapter" is useful too.

Avoid over-subdivision of your document. By this I mean don't subdivide where it is not needed. If the subject you are referring to has a sufficient description in one or two sentences then don't put it into a separate document, but include it straight into the text. The user doesn't want to have

to keep moving between documents for what may be relatively trivial descriptions; this interrupts the continuity of the text and slows down the rate of information gathering. However, there is a case where small documents are excusable, and that is in the situation of maintaining consistency. If you are describing a set of similar objects (say gadgets in a window) and some descriptions are long and others are small, it is better to be consistent and put the small descriptions in their own document as well as the long ones.

Consistency

Consistency is important in your database and it's easier to forget it when writing AmigaGuide databases because, unlike writing continuous texts, you don't have all your previous work in front of you. So you'll need to make the effort to enforce consistency. A good idea is to use templates for documents if you are constructing many documents about similar things (for instance, see the descriptions of

ARexx~commands
in this database).

It is also important to be consistent in your use of links. If there are, say, three instances of a link-phrase (that is, a phrase which has a related document giving more information about it) in a document, do not just make links out of two of them; acceptable strategies are either only making a link from the first instance, or preferably make a link out of each of them. That, I guess, is a matter of taste.

Layout

The biggest problem with the layout of AmigaGuide databases is the disparity between AmigaGuide v39 and AmigaGuide v34. As AmigaDOS 2.04 is still a "current" operating system and by no means obsolete, it is very important to continue supporting those users by not making the database inaccessible to them, unless you can guarantee that only AmigaDOS 3.x+ users will be reading the file. This means, effectively, not using the AmigaGuide wordwrap option and relying on Heddley's formatting. That is a must otherwise your files will be unreadable with AG V34.

Paragraphs

Heddley operates very much with the concept of paragraphs. They are an inordinately sensible way of organising text and one that should come naturally to most. A paragraph in Heddley is a block of text terminated with a blank line, or a line with indented text. A paragraph is considered fair game for reformatting and so it is irrelevant where you place the newlines when first inputting the text, as long as you terminate the paragraph properly.

An indented line will not be joined to a preceding line, this enables you to create lists and tables, simply by indenting each line by one or more space. This may seem a strange way of doing things, but it is a sensible method without having to invent any other control information to tell Heddley what to do with the text. This also means that you may indent the leading line of your paragraphs with however many spaces you like. Be aware though that you can only indent the first line, otherwise Heddley will not treat your block of text as a paragraph, but as a list instead.

It may be that you are used to writing your text with a text editor in a format that breaks the rules that Heddley uses. I'm afraid that you'll just

have to change the way you do things to use Heddley. I have chosen a reasoned and sensible method of handling text that provides full support for AG V34 users too. Writing AmigaGuide databases is a different kind of activity to writing plain text files, and you will have to do things differently. AmigaGuide databases that are basically chopped-up text files will not read anything like as well as an AmigaGuide database written to be one.

Headings

A lot of people seem to write headings like this:

```
My Heading
=====
```

While this obviously makes the heading clear for AG V34 users who have no underline, bold etc. it is bad style and old fashioned. For a start, headings like this that are flush left will be treated as a paragraph and thus turn into:

```
My Heading =====
```

I recommend that you use the underline facility in AG V39 (available from the Style window) for headings. You may prefer to use bold. For AG V34 users your headings will simply appear as separate lines, which is normally good enough to indicate a heading anyway. Please don't use symbols to form underlines - it just isn't nice!

At the moment Heddley turns underlining into typewriter text when converting to HTML. HTML does not have underline but I wanted to give access to the typewriter font. If you plan to convert to HTML I recommend using bold for headings rather than underline, as this will be reproduced faithfully.

Fonts

Again the use of fonts brings up the disparity between AG V34 and AG V39. It makes no sense at all to use proportionally spaced fonts without using the V39 wordwrap option, or your text will look ugly. So my recommendation is that you only change the font for V39 databases. My personal opinion is that the user should be free to choose what font they want to read the text in, and the documentation writer should just be able to change the style (bold, italic etc). If you want to write a document with full font control, use a word processor!

Emphasis and color

Traditionally, italicised text is used for emphasising words or phrases. However I think it is acceptable to use bold in AmigaGuide documents (probably it is not a good idea to use bold if you are already using it for headings), largely because of the poor readability of the Amiga's algorithmic italicised fonts, especially with small bitmapped fonts.

I'm not yet sure what place to give color in AmigaGuide documents. HTML supports no color so it would be foolish to rely on color. The most useful application I can think of color is syntax coloring. That is, coloring phrases or words that have similar purposes in the text the same distinctive color. For example, you may want to color all text that is meant to represent user input in one particular color (in the absence of a set typewriter style) to

mark it out as being user input.

A final word on emphasis and color: be sparing. Overuse of these techniques will lessen their impact on the reader. The focus should always be the text, so that the reader can understand what you have written more readily.

Multimedia

With the advent of datatypes, there is great scope for including pictures and sound as part of your AmigaGuide databases. Indeed for documentation it would be wonderful to be able to include a screenshot of the window you are describing instead of resorting to painting poor ASCII pictures. The only problem with this (and it's a big problem) is that again V34 owners lose out. If you can be bothered to go to the trouble, try using an external viewer such as "WDisplay", distributed with the AmigaGuide distribution on Aminet. Making buttons which are AmigaDOS calls to WDisplay will work for both V34 and V39. Otherwise make it plain in the documentation that V34 owners will not be able to see the pictures; it is confusing for users to find an inoperative button, and reflects badly on you. Let's just hope that everyone with V34 will upgrade soon (and can afford it!)

Document naming

For your own convenience, it is a good idea to give your documents IDs which convey immediately to you their contents. For instance if you are documenting the menu items of a program you could name the documents "menuitem_load", "menuitem_save" and so on... These could be grouped in the listview, and thus will make everything easier for you. Naming your documents "XX1", "XX2" and so on is not a good idea.

1.73 Save as ASCII

This option saves the database as in
Save~as~ANSI
, except it removes the ANSI
color codes and saves just plain ASCII text.

1.74 Index

Index of database Heddley

Documents

About the author

ADD

AmigaGuide ARexx Port

AmigaGuide V39+

ARexx examples

ATTRIB

Change text styles

Clear text

Configuring Heddley

DELETE

Delete button

Distribution of Heddley

Document Pick List

EDIT

Edit text

Editing a document

Editing the database

GOTO

Heddley v1.1 (C) Edd Dumbill 1994

How do I?

INFO

Installing Heddley

Introduction to Heddley

LOAD

Load database

Load text

LOCK

Make into button

MOVE

Moving a document - Down

Moving a document - Up

NEW

On line help

Other Information

PENS

PREFS

QUERY

Quick start guide to using Heddley

QUIT

Replicate button

Save

SAVE

Save as ANSI

Save as ASCII

Saving HTML source

Smooth scroll

Starting a new database

Starting Heddley

TEST

Test Database

The Add document gadget

The ARexx port

The Attrib gadget

The BGPEN tooltype

The Button Edit window

The Current Pen cucle gadget

The Delete gadget

The document name list

The EDITOR tooltype

The FGPEN tooltype

The Font Button

The FONT tooltype

The FONTSIZE tooltype

The HELPFILE tooltype

The Info Window

The Palette Gadget

The Preferences Window

The TEXTWIDTH tooltype

Things I plan to implement in Heddley

Version information

What About...?

Why should I use Heddley?

Writing Amigaguide files

Buttons

~ADD~~~~~

~ARexx~example~scripts~~~~~

~ATTRIB~~

~Configuring~Heddley~~~~~

~DELETE~~

~EDIT~~~~~

~Editing~a~database~~~~~

~Feature~list~~~~~

~GOTO~~~~~

~How~do~I~do...?~~~~~

~How~to~contact~the~author~

~HTML~output~~~~~

~IMPORTANT~---~DISTRIBUTION~

~Index~~~~~

~INFO~~~~~

~Installing~Heddley~~~~~

~Introduction~~~~~

~LOAD~~~~

~LOCK~~~~

~MOVE~~~~

~NEW~~~~

~On-line~help~function~~~~

~Other~information~~~~~

~PENS~~~~

~PREFS~~~

~QUERY~~~

~Quick-start~guide~~~~~

~QUIT~~~~

~SAVE~~~~

~Starting~Heddley~~~~~

~Style~and~things~~~~~

~TEST~~~~

~The~ARexx~port~~~~~

~To-do~feature~list~~~~~

~Version~information~~~~~

~What~about...~?~~~~~

'Dn'

'Prefs...'

'Test~Database'

'to-do'

'Up'

Add

AmigaGuide~AREXX~port

AmigaGuide~V39

AmigaGuide@~V39

Apply~style~to~text

ARexx

ARexx~commands

ARexx~port

ARexx~scripts

Attrib

ATTRIB

Attrib~Window

author

auto-index

automatic~index~creation

BGPEN

BGPEN~tooltype

button~edit

Button~Edit~window

BUTTPEN~tooltype

Clear~text

click~here

click~here

click~here

click~here

click~here

click~here

CLICK~HERE

click~here

click~here

click~here

Contents

contents

Current~pen

Database~documents

databaseinfo.rexx

Delete

Delete~button

Dn

document~edit~window

document~name~gadget

Edit

Edit~Attribute

Edit~attributes

Edit~text

edit~window

editing~a~document's
editnode.rexx

EDITOR

Editor~string

EDITOR~tooltype

examples

FGPEN

FGPEN~tooltype

FONT

Font

Font~size

FONT~tooltype

FONTSIZE

FONTSIZE~tooltype

GOTO

hedd_empty.rexx

hedd_golded_temp.rexx

Help

HELPPFILE

index

Info

INFO

info~window
Load
Load~new~text
main~listview
Make~into~button
me
New
normal~AmigaGuide~help~file
on-line~help
Palette
PENS
preferences~window
Prefs...
QUERY
Replicate~button
Save
Save~as~ANSI
Save~as~ASCII
Save~as~HTML
savenode.rexx
Saving~ANSI~source
Saving~HTML~source
Shell
Smooth~scroll
Test~Database
Text~width
TEXTWIDTH
TEXTWIDTH~tooltype
tooltypes

Up

window