

ADtoHT

COLLABORATORS

	<i>TITLE :</i> ADtoHT		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 26, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ADtoHT	1
1.1	ADtoVT V1.3	1
1.2	background	1
1.3	legal	2
1.4	warning	3
1.5	usage	3
1.6	notes	5
1.7	cygnused	6
1.8	thanks	6
1.9	history	6
1.10	todo	7

Chapter 1

ADtoHT

1.1 ADtoVT V1.3

ADtoHT V1.3
12-Aug-93

A program to convert AutoDoc files to hypertext format

What is it

Legal Matters

WARNING

Using ADtoHT

Notes

Expansion of 'typedef struct' expressions

'Short' AutoDocs

Reading crossreference files

Thanks

History

ToDo

1.2 background

ADtoHT V1.3 Background

=====

Are you a programmer? If not, you won't need this program.

Ever experienced a situation like this: You want to find out how to do xy. You pull out your favorite AutoDoc

retrieval program, find references to other functions and/or include files, and so you keep typing and typing and typing until know what you were looking for. Assuming that you want to program instead of practicing your typing-skills, here is the solution: ADtoHT.

This program will convert your AutoDoc files to AmigaGuide hypertext format. It will create links within an AutoDoc file, to other AutoDoc files, and to your include files.

I know that Commodore distributes a similar program called AD2HT. However, a friend of mine told me that it took him quite some time and several ARExx scripts to convert the autodocs into AmigaGuide files, so I decided to write my own ADtoHT. I never tried AD2HT, so I don't know whether ADtoHT is really better. You decide.

1.3 legal

ADtoHT V1.3 Legal Matters =====

ADtoHT V1.3 is © 1993 by Christian Stieber

I cannot be made responsible for anything that happens directly or indirectly through the use of ADtoHT. If it crashes your system, causing you to loose vital data; if it blows up your harddisk, turns your joystick into a flying saucer or kills your grandma:

```
*****  
* DON'T BLAME ME. IT'S NOT MY PROBLEM. *  
*           USE AT YOUR OWN RISK!           *  
*****
```

ADtoHT V1.3 is FREeware. It may be spread freely, as long as all files (a list of all files is given below) are kept together unmodified.

This does not apply to commercial packages. If you really want to include ADtoHT with a commercial package, a shareware package, or some form of bookware, you must get my written permission first.

You may not spread modified versions of ADtoHT.

Christian Stieber
Konradstraße 41
W-8070 Ingolstadt
Germany

EMail while I'm still a student:

stieber@informatik.tu-muenchen.de

Any distribution of ADtoHT V1.3 must include the following files:

ADtoHT	the executable
ADtoHT.guide	this documentation file
adfile.c	the sources, using SAS/C V5.10
adfile.h	
convertad.c	
convertad.h	
incfile.c	
incfile.h	
main.c	
misc.c	
misc.h	
global.h	
readdir.c	
readdir.h	
shortad.c	
shortad.h	
readxref.c	
readxref.h	
MyStartup.h	
MyLib.h	
MyStartup.o	
MyLib.lib	
lmkfile	

1.4 warning

A WORD OF WARNING

=====

ADtoHT V1.3 is basically a collection of hacks. As such, it may display strange behavior if YOU do things differently than I. Please tell me if you think something is wrong, and I'll try to fix it.

I didn't have any real concept when writing ADtoHT, so the structure of the program isn't really thought out -- you know, insert a flag here, do some unexpected side-effect there... it even has gotos in it's code.

Also, feel free to tell me what other features you think may be useful. If I consider them useful, I might even implement them...

Also, speed was not a really important issue. It does use binary searches, but that's about all I did to speed it up. A lot could be done in this area, but ADtoHT is not a program you use every day, so it doesn't seem very important to me. If you want more speed, get a 68060, or write your own ADtoHT.

1.5 usage

Using ADtoHT V1.3

=====

ADtoHT requires Kickstart V37 or later.

ADtoHT is a shell-only program. If you try to run it from Workbench it will exit quietly.

ADtoHT has the following template:

```
AUTODOCS/A, INCLUDES/A, HYPERTEXT/A, XREF/K, ERROR/K,  
EXTENSION/K, NOEXPAND/S, TYPES/S, CED/S
```

AUTODOCS/A: This is the directory where your AutoDoc files are located.

INCLUDES/A: This is the directory where your system include files are located.

HYPERTEXT/A: This is the directory where you want your hypertext files to be written.

XREFOUT/K: This is where the crossreference file should be written. The default is "autodoc.xref" in your current directory. Note that a crossreference file will always be written; if you don't need it, use XREFOUT=NIL:

XREFIN/K: This specifies a crossreference to be read. Defaults to no file. See Notes for more information.

ERROR/K: An optional error file. All warnings will be printed on the CONSOLE: and written to this file, so you don't need to watch your shell-window all the time. Defaults to no error file.

EXTENSION/K: The extension for your hypertext files. When creating the filenames, the ".doc" will be removed from the AutoDoc filename, and the new extension will be appended. Defaults to ".guide".

NOEXPAND/S: Prevents expansion of "typedef struct". See Notes for more information on typedef struct expansion.

TYPES/S: By default include files named "exec/types.h" will not be processed. This prevents links to types such as WORD, ULONG etc. Furthermore, typedefs defining "BSTR" and "BPTR" are ignored. If this switch is used, "exec/types.h" files will be processed normally, and "BSTR"/"BPTR" typedefs will not be ignored.

CED/S: This affects the line numbering scheme for warnings/errors. CygnusEd handles FormFeed characters as newlines, so this switch makes sure that the line numbers given by ADtoHT are "CygnusEd-compatible".

SHORTTOC/S: By default, the table of contents of a hypertext file created by ADtoHT will look like this:

```
exec.library/AbortIO()
exec.library/AddDevice()
exec.library/AddHead()
...
```

With SHORTTOC, only the function names will be shown:

```
AbortIO()
AddDevice()
AddHead()
...
```

1.6 notes

ADtoHT V1.3 Notes

=====

1) Expansion of "typedef struct" expressions:

The easiest way to explain this is by using an example. The <libraries/commodities.h> file defines the following:

```
typedef struct InputXpression IX;
```

If ADtoHT detects a reference to IX while converting AutoDoc files to hypertext, it will not make a link to the line containing the "typedef" statement, but it will create a link to the definition of "struct InputXpression", which is usually what you are really interested in.

This expansion is performed for all "typedef struct ... x;" statements, unless the command line switch NOEXPAND is used.

2) "Short" AutoDocs

There are two ways to reference another function: the "long" form, e.g. "exec.library/AllocMem", and the "short" form, e.g. "AllocMem". There are a few functions that have more than one description, e.g. OpenDevice(), AbortIO() etc. To resolve a "short" AutoDoc, ADtoHT will first check the current file, then it will try libraries, devices, datatypes, libs and resources (in that order). For example: ADtoHT is processing timer.doc and finds a reference to "OpenDevice". It will first look for a timer.device/OpenDevice (which doesn't exist), then for a #?.library/OpenDevice, finding exec.library/OpenDevice.

3) Reading crossreference files

This option allows you to update your AutoDocs or add additional AutoDocs. Using XREFIN causes ADtoHT to read a crossreference file, which MUST have been written by ADtoHT (ADtoHT makes quite a few assumptions about the structure of this file). However, you should be careful when using this option: If your current Includes and AutoDocs do not have unresolved references to the new files (this is generally true for additional libraries from sources other than Commodore, e.g. arp.library, req.library, toolmanager.library etc) this option is safe to use. If you have new Includes, you should not use this option, unless you know that line numbers have not changed and that there are no new structures and typedefs. If you get a new AutoDoc file which replaces an existing one, you may use this option if there are no new functions within the new file.

Here's how to update existing or add new AutoDocs:

- 1) Create a directory that contains the new AutoDocs ONLY.
- 2) Create a directory that contains the new Includes ONLY.
- 3) Use ADtoHT on the above directories, reading the old crossreference file.
- 4) The directories created in steps 1) and 2) may now be deleted, as long as the include files are kept within the AmigaGuide search path.

1.7 cygnused

CygnusEd: an editor by CygnusSoft Software
published by ASDG Inc.

1.8 thanks

Thanks must go to...

Christine	for being such a great person
Hans Maurer	for many suggestions and ideas
Commodore	for V37++. Do we finally get RTG in OS 4.0?
SAS Institute	for the compiler
CygnusSoft Software	for CygnusEd
Lots of people	for great FD/PD/Shareware/Whatever software

and of course to all people giving me feedback about ADtoHT.

1.9 history

V0.0 (23.01.93) First release.

V0.1 (03.02.93) Major bug fixed, XREF-File is now usable

Code reordering allows XREFIN and XREFOUT to be the same file. Thanks to Hans Maurer for the suggestion.

- V1.0 (11.03.93) Errorfile is now optional, as documented. Thanks to Hans Maurer for reporting the bug. First public release.
- V1.01 (30.03.93) Fixed typo that prevented reading directory trees that are more than 2 levels deep. Fixed bug that corrupted the memory list. Really should use MungWall for debugging... Fixed bug that prevented ADtoHT from reading it's crossreference files. Changed the required format of the first line in an autodoc entry. May now be xxx.yyy xxx.yyy or just xxx.yyy
- V1.1 (30.04.93) Actually, the crossreference wasn't usable as intended, since the names of the nodes were different from the names that appeared in the crossreference file. This is now fixed; if you still need the old names in addition to the new ones, use LONGNAMES. Doesn't require FormFeeds at the end of an autodoc file. Hopefully this doesn't introduce a bunch of new bugs...
- V1.2 (25.05.93) Added switch SHORTTOC requested by Jörg Bublath. Not released.
- V1.2 (04.08.93) Finally fixed the nasty thing with ADtoHT complaining about missing spaces/tabs in autodoc node headers... Almost everybody who gave me feedback suggested that this should be fixed, so finally I did it...
- V1.3 (12.08.93) Fixed bug that kept in because the XREF file changed so many times: ADtoHT wasn't able to read it's XREF files anymore. Removed braindamaged LONGNAMES switch. If function Name() is not found, it will try to find NameA(), NameTags() and NameTagList() instead. Please don't complain about speed... this was the easiest way to implement the recognition of varargs stubs.

1.10 todo

Things to expect from future version of ADtoHT

=====

Planned for V1.4:

- * Create an index for structures and functions in alphabetical order

Planned for V1.5 - V1.9:

No plans yet -:)

Planned for V2.0:

Use of a "real parser" to process the includes. This should give me the ability to create hypertext versions of the include-files as well, i.e. all instances of "struct Window" will become links to the definition of "struct Window".

And of course, it will clean up the code -- after all, ADtoHT started as a hack.

It will also solve the problem of finding out whether something is a function() or a #define. Currently, ADtoHT thinks that all MUI methods and attributes are functions, since they have lowercase letters in them.
