adoc

COLLABORATORS

	TITLE : adoc		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		December 7, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

adoc

1.1 adoc.guide

```
     Warning: The ADOC project and this documentation are far from complete!
              Please report bugs and direct comments to

                       Tobias Ferber, <ferber@rpk.mach.uni-karlsruhe.de>
```

This document describes ‘ADOC’ version 1.01, which extracts Autodocs or Texinfo from source code.

Overview & Legal stuff

Controling ‘ADOC’

What you should be aware of

Appendix

1.2 adoc.guide/Copying

GNU GENERAL PUBLIC LICENSE

****

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

2. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

3. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

4. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

5. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

6. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

7. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

8. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

9. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

10. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

11. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

12. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

13. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR

```
     INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
     DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU
     OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY
     OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN
     ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

                      END OF TERMS AND CONDITIONS
```

How to Apply These Terms to Your New Programs
===

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
     ONE LINE TO GIVE THE PROGRAM'S NAME AND AN IDEA OF WHAT IT DOES.
     Copyright (C) 19YY  NAME OF AUTHOR

     This program is free software; you can redistribute it and/or
     modify it under the terms of the GNU General Public License
     as published by the Free Software Foundation; either version 2
     of the License, or (at your option) any later version.

     This program is distributed in the hope that it will be useful,
     but WITHOUT ANY WARRANTY; without even the implied warranty of
     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
     GNU General Public License for more details.

     You should have received a copy of the GNU General Public License
     along with this program; if not, write to the Free Software
     Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
     Gnomovision version 69, Copyright (C) 19YY NAME OF AUTHOR
     Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
     type `show w'.  This is free software, and you are welcome
     to redistribute it under certain conditions; type `show c'
     for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
     Yoyodyne, Inc., hereby disclaims all copyright
     interest in the program `Gnomovision'
     (which makes passes at compilers) written
     by James Hacker.

     SIGNATURE OF TY COON, 1 April 1989
     Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.3 adoc.guide/Contributors

Contributors
****

- Tobias Ferber, `<ukjg@rz.uni-karlsruhe.de>' wrote `ADOC' and this documentation.

- Harald Kunze, `<kunze@rpk.mach.uni-karlsruhe.de>' contributed most of the ideas and forced the project to come into life.

1.4 adoc.guide/Introduction

Introduction
****

The `ADOC' scanner creates Autodocs or Texinfo source from source code. It is an enhanced version of Bill Koesters `Autodoc' program as of 21-Aug-89 which is part of C=ommodore's Native Developer Kit (NDK).

`ADOC' scans the input files for a special kind of comments: the autodoc comments. Autodoc comments start at the begining of a line with a string of the form

```
     ?***??* lib/fun ...
```

The `?' stands for characters which are ignored, they must however be "black", i.e. they must not be tabs or spaces. The `*' ist ment literally, so in these places there must be an asterisk. `lib/fun' is the identifier which specifies the library and the name of the function described in the following comment. `lib' and `fun' consist of upper- and/or lower case letters, decimal digits, a dash `-' and/or the underscore `_'. The rest of the line is ignored.

E.g.: A standard C language autodoc comment begins like this:

```
     /****** mylib/myfun **************************
```

The following lines make up the function description, a sequence of one or more sections. A section begins with the section heading, a line with only -- and at least two -- capital letters. Some common section headings are

NAME, SYNOPSIS, FUNCTION, INPUTS, RESULT, ...

Each line in the function description must be prefixed with a black character followed by at least one white space or the end of the line. An autodoc comment ends with a line beginning with a black character followed by at least two asterisks ‘*’. An example template comment can be found at the end of this document. See Style Guide, for further details.

1.5 adoc.guide/Options

Options

‘ADOC’ reads the command line from left to right and collects filenames and options before it starts processing your input. The filenames are queued and processed in the same order as they appear in the command line. The list-file option ‘@’ is replaced by the contents of the list file read in top-down order. If no input files are specified ‘ADOC’ will read from the standard input stream.

Autodoc Options	‘Autodoc’ output only
IO Options	both ‘Autodoc’ and ‘Texinfo’ output
Texinfo Options	‘Texinfo’ output only
Warning Options	influencing ‘ADOC’’s pedanticness
Miscellaneous Options	everything else

1.6 adoc.guide/Autodoc Options

Options for the ‘Autodoc’ output
================================

‘--autodoc’

‘ADOC’ creates an autodoc file. (This is the default)

‘--no-form-feeds’

Usually the functions in ‘ADOC’s autodoc output are seperated by form feeds (‘^L’). With this switch ‘ADOC’ omits them. (This switch is a no-op in ‘--texinfo’ mode).

‘--table-of-contents’

A table of contents is written to the output followed by the autodoc functions. Functions and toc-entries are always in the same order, so ‘--preserve-order’ also affects the table of contents. (This switch is a no-op in ‘--texinfo’ mode).

‘--preserve-order’

With this option ‘ADOC’ does not sort the functions before writing the autodoc output. (This switch is ignored in ‘--texinfo’ mode).

1.7 adoc.guide/IO Options

Options for ‘Autodoc’ and ‘Texinfo’ output

==

‘--stdin’

This switch behaves like a filename for the standard input stream. Using it does not only force reading from stdin but you can also specify the files to read before and after the standard input.

‘--output-file FILENAME’

This option allows you to specify a filename for ‘ADOC’’s output. If this option is ommited ‘ADOC’ will write to the standard output stream.

‘--error-file FILENAME’

Like ‘--output-file’ this option allows you to redirect ‘ADOC’’s output. The difference between the two is that this option redirects warnings and error messages from stderr into the file FILENAME.

‘--flip-slashes’

This option converts ‘*’ to ‘/*’ and ‘*\’ to ‘*/’. This allows C language comments in the autodocs without nesting comments in the source code.

The name of the option might give you the idea of a general conversion of backslashes into slashes. This is however incorrect. The character on the left or right of the ‘\’ must be an asterisk (‘*’). Please note that ‘**’ is transformed into ‘*/*’.

‘--tabs-to-spaces’

Tabs in the input are turned into spaces with a default tab size of 8. This is the default in ‘--texinfo’ mode.

‘--tabsize NUMBER’

The tab size is changed to NUMBER. (default is ‘--tabsize 8’). This option implies ‘--tabs-to-spaces’.

‘--line-length NUMBER’

Function headers in the autodoc output will be NUMBER columns wide. (default is ‘--line-length 80’). (This option has no influence on ‘ADOC’’s Texinfo output).

‘--internal’

‘ADOC’ will only read comments with an ‘i’ in column 5 or 6. (same as ‘--yank-type "i"’).

‘--yank-type TYPE’

‘ADOC’ will only read comments which have in column 5 or 6 one of the chatacters in the TYPE string.

‘--indented-comments’

The autodoc comments normally have to start at column 1, i.e.: they must not be indented. This option allows indented comments, but it does not check whether or not the indentation of all comment lines is equal or not.

Additionally, if the indentation is not a multiple of ‘ADOC’’s tab size, the conversion of TAB characters into spaces might lead to wrong results. This is due to the fact that ‘ADOC’ does not remember the columnar position of each body text line but silently assumes that they all start at a tab stop column.

In general it is *not* recommended to make use of TABs in indented comments. If you really insist on them you can pipe these files through ‘expand’ and ‘ADOC’’s output through ‘unexpand’:

```
$ cat *.c | expand -t8 | adoc -I | unexpand -t8 -a > adoc.out
```

‘--reindent-bodytext NUMBER’

This option changes the indentation of lines in the body text of all functions to NUMBER spaces. If NUMBER is 0 then the current indentation is removed which is default in ‘--texinfo’ mode.

In order to remove the current indentation ‘ADOC’ cuts a columnar block beginning at column 1 from the function body. The width of this block is the maximum width without hurting any text.

There is one more thing to mention: This option automatically converts all TAB characters into the apropriate number of spaces. You want this because the new indentation might destroy your textual layout otherwise. The conversion is performed before the removal of the indentation.

If you want to change the body text indentation via ‘--reindent-bodytext’ but insist on having TAB characters instead of spaces then you have to pipe the ‘ADOC’ output through ‘unexpand’:

```
$ adoc -I *.c | unexpand -t8 -a > adoc.out
```

‘--preserve-indentation’

This option avoids reindentation of the body text (e.g.: in ‘--texinfo’ mode where ‘--reindent-bodytext 0’ is ‘ADOC’’s default).

‘--sort-entries’

This option forces sorting of the functions which is default in both ‘--autodoc’ and ‘--texinfo’ mode.

‘-MMACRO VALUE’

Defines a value VALUE for a macro MACRO which is expanded when writing the body text of a function. (e.g.: the above mentioned conversions of for ‘--flip-slashes’ and ‘--texinfo’ are implemented in this manner).

Note that there must *not* be a space between ‘-M’ and the name (i.e. the source string) of the macro.

‘-U MACRO’

Undefines a macro MACRO no matter whether MACRO had been defined with ‘-D’ or ‘-M’ or not at all.

1.8 adoc.guide/Texinfo Options

Options for ‘Texinfo’ output
============================

‘--texinfo’

Write texinfo source instead of autodocs. This switch implies ‘--tabs-to-spaces’. ‘--preserve-order’ and ‘--no-form-feeds’ are ignored. The functions in the texinfo output are always sorted, tabs are always expanded and there will be no form feeds.

Some conversions will be performed on your autodoc body text, especially:

```
     @  ->  @@
     {  ->  @\{
     }  ->  @}
```

‘--project’
‘--revision’
‘--edition’
‘--copyright’
‘--author’

‘ADOC’’s texinfo header reserves these four macros. Their value defaults to their name. ‘--project’ is not only used on the title page but also as the name for the ‘.info’ or ‘.guide’ file.

These options actually define the macros ‘PROJECT’, ‘EDTITION’, ‘REVISION’, ‘COPYRIGHT’ and ‘AUTHOR’ as if they had been defined via ‘-D’.

‘--no-references’

In ‘--texinfo’ mode, ‘ADOC’ tries to find matching nodes for the identifiers in the ‘SEE ALSO’ section. You can however force ‘ADOC’ to write this section as it is by giving this switch.

Best results can be obtained by writing one reference per line in the same ‘lib/fun’ format as in the header line of the referenced function. ‘ADOC’ will however try to prepend the current functions library name to the identifiers and removes parens, commata, white spaces and newline characters in order to find the referenced ‘lib/fun’ header.

‘--parse-references’

This option prints one "See ..." sentence containing all references from the ‘SEE ALSO’ section. No itemization is used.

The name of this option might insinuate the necessity of passing it together with other options which need to analyze the ‘SEE ALSO’ section. This is however incorrect. The name of this option is somehow an archaic requilict from those times where ‘ADOC’ learned to look closer at it’s input. It could also be called ‘--make-one-sentence-of-references’ or something similar.

‘--itemize-references’

‘ADOC’ puts the parsed references as an ‘@item’ into an ‘@itemize’ environment. (This is default in ‘--texinfo’ mode).

Parsing the references is done automatically, even if you guess that you need the ‘--parse-references’ option. This not a good guess. Perhaps you should better look at the above description of this option.

‘--no-grouping’

‘ADOC’ will not make use of texinfo’s grouping feature. (This is the default).

‘--group-sections’

The section body (following the title keyword) will be grouped into a ‘@group’ ... ‘@end group’ environment in order to prevent TeX from breaking the page inside a section. The current implementation of ‘texinfo’ does not allow placing the section keyword into the ‘@group’ as well:

```
! @group invalid in context where filling is enabled.
```

‘--body-environment STRING’

This option allows you to specify the texinfo environment into which the section body text will be placed. The internal default is ‘--body-environment smallexample’ and so each section body is placed into an environment of the form

```
@smallexample
  <body text>
@end smallexample
```

If ‘--group-sections’ is specified, then this ‘@group’ is placed into the given environment like this:

```
@smallexample
@group
  <body text>
@end group
@end smallexample
```

Please note that this environment is not used within parsed references in the ‘SEE ALSO’ section.

‘--texi-funtable’
This switch tells the texinfo code generator to place a function into a ‘@table’ environment with each section keyword as a ‘@b{}’old ‘@item’. This might however look ugly and lead to an immense amount of overfull ‘\hbox{}’es because TeX will then indent the following body text twice (for the ‘@item’ and for the ‘@example’) and so ‘ADOC’’s default is to write a section keyword like an ordinary, indented, ‘@b{}’old paragraph.

‘--form-feeds’
With this option a page break is forced by inserting a ‘@page’ command in front of each function description. An exception is the first function of a library which always begins on the same page as the library ‘@chapter’.

‘--texi-header-file FILENAME’
‘ADOC’ comes up with a minimal header and titlepage when writing texinfo output. This option allows you to replace ‘ADOC’’s internal header with the contents of FILENAME.

Note that a macro expansion of macros defined via ‘-D’ will be performed on the contents of FILENAME.

‘--no-texi-header’
This option forces ‘ADOC’ to ommit writing a texinfo header and a title page. Texinfo output begins with the (Top) ‘@menu’.

‘--no-texi-index’
No function index and no table of contents is used in the texinfo file.

‘-DMACRO VALUE’
Defines a value VALUE for a macro MACRO which is expanded when writing the texinfo header. These definitions are *not* performed on the body text of a function description like those defined via ‘-M’.

Note that there must *not* be a space between ‘-D’ and the name of the macro.

1.9 adoc.guide/Miscellaneous Options

Miscellaneous Options
=====================

‘--no-output’
‘--dry-run’
‘ADOC’ only reads your input (and eventually prints warnings). No output will be written.

‘--silent’
‘ADOC’ will behave more briefly with this option.

‘--help’

A list of options and a short description will be printed to the standard output stream.

If your terminal is limited to 80 columns then you can use

‘-h0’
To print only the ‘usage:...’ line

‘-h1’
To print only the "short" options (those with one dash ‘-’).

‘-h2’
To print only the "long" options (those with two dashes ‘--’).

‘--version’
Author and version information will be printed.

1.10 adoc.guide/Warning Options

Influencing ‘ADOC’’s pedanticness
=================================

‘--no-warnings’
‘ADOC’ will not print *any* warnings.

‘--check-keywords’
‘ADOC’ will complain about section title keywords which are not in the example below.

‘--check-presence’
‘ADOC’ will complain if one of the keywords in the example below is missing in a function description.

‘--pedantic’
‘ADOC’ will complain about almost everything.

1.11 adoc.guide/Style Guide

Style Guide
****

This chapter is under development.

How ‘ADOC’ reads your input
===========================

‘ADOC’ scanns your input for autodoc comments. These comments begin on the leftmost column in a new line which most likely looks like one of those:

‘/****** mylib/myfun ******’
     This is the beginning of a standard C or Rexx Language autodoc
     comment for the function ‘myfun’ in the library or module ‘mylib’.

‘;****** mylib/myfun ******’
‘******* mylib/myfun ******’
     Most assemblers use a semicolon (‘;’) or an asterisk (‘*’) to
     begin a remark.

‘#****** mylib/myfun ******’
     This type is known in Makefiles and shell scripts.

‘%****** mylib/myfun ******’
     For autodoc comments in TeX or ‘METAFONT’ we need the percent sign
     (‘%’).

When ‘ADOC’ encounters the beginning of an autodoc comment it starts collecting section headings and body text lines and saves them for later use in the code generator.

The lines inside autodoc comments must be prefixed by a black character. This character is made invisible in the output. To be more precise: ‘ADOC’ removes the black character in column 1 if the following character is a TAB or a newline character but it replaces it by a space if the following character is a space character.

Each line which contains entirely capitalized letters is assumed to be a section heading unless it contains only one single upper case character.

The indentation of section headings and body text lines does not matter. You should however use an indentation of 3 spaces for a section heading and 7 spaces for body text lines. Note that ‘ADOC’ allows you to adjust the indentation for the output via the ‘--reindent-bodytext’ option.

Example comment

Here is a sample "correct" autodoc comment taken from the ‘autodoc.style’ which is part of Commodore’s Native Developer Kit (NDK):

```
/****** financial.library/StealMoney ****************************************
*
*  NAME
*     StealMoney -- Steal money from the Federal Reserve Bank. (V77)
*
*  SYNOPSIS
*     error = StealMoney( userName,amount,destAccount,falseTrail )
*     D0,Z                D0       D1.W    A0          [A1]
*
*     BYTE StealMoney
*          ( STRPTR,UWORD,struct AccountSpec *,struct falseTrail *);
*
*  FUNCTION
*     Transfer money from the Federal Reserve Bank into the specified
*     interest-earning checking account.  No records of the transaction
*     will be retained.
*
*  INPUTS
*     userName      - name to make the transaction under.
*                     Popular favorites include "Ronald Reagan"
*                     and "Mohamar Quadaffi".
*     amount        - Number of dollars to transfer (in thousands).
*     destAccount   - A filled-in AccountSpec structure detailing the
*                     destination account (see financial/accounts.h).
*                     If NULL, a second Great Depression will be triggered.
*     falseTrail    - If the DA_FALSETRAIL bit is set in the destAccount,
*                     a falseTrail structure must be provided.
*
*  RESULT
*     error - zero for success, else an error code is returned (see
*             financial/errors.h).  The Z condition code is guaranteed.
*
*  EXAMPLE
*     Federal regulations prohibit a demonstration of this function.
*
*  NOTES
*     Do not run on Tuesdays!
*
*  BUGS
*     Before V88, this function would occasionally print the address and
*     home phone number of the caller on local police 976 terminals.
*     We are confident that this problem has been resolved.
*
*  SEE ALSO
*     CreateAccountSpec(),security.device/SCMD_DESTROY_EVIDENCE,
*     financial/misc.h
*
****************************************************************************
*
*  Private notes:
*     A4=stringbean
*     A3=dogbreath
*  Must preserve A1 for taxshelter.library
*/
```

There are three spaces from the '*' to the start of each 'HEADING'. There is one tab after the '*' before the start of the body text. No other tabs are used.

Note that 'ADOC' also allows the use of spaces to indent body text lines. You can even mix tabs and spaces! (We recommend to avoid TAB characters.)

General style notes

- It is critical that changes be noted in a sane manner.

- When referring to a function, the standard format is 'FunctionName()'.

- Capitalization should be correct. Here are some guidelines:

1. Trademarks, *must* be capitalized.

2. Names of "things" are as defined. For example, ‘OpenWindow()’, and "a ‘Window’ structure". "fiddles with your window" does not refer to the structure, and should not be capitalized.

* Lines terminate at column 79, so the autodocs are readable from within an 80 column window.

For further guidelines refer to ‘autodoc.style’ -- Commodore’s autodoc style guide which is part of the Native Developer Kit (NDK).

1.12 adoc.guide/Bugs

Bugs
****

- Most of ‘ADOC’’s "short" options are compatible with Bill Koester’s ‘Autodoc’ Program. ‘-C’, ‘-s’, ‘-a’, ‘-F’, ‘-r’ and ‘-w’ are ignored for compatibility.

- This documentation is far from being complete

- Most of the options are not tested

- The ‘-M’ option actually does not affect the ‘SEE ALSO’ section in texinfo source if parsing is enabled. This might change in the future when using ‘funexpand()’.

- TeX will not break a page inside of indented paragraphs in ‘@example’ or ‘@smallexample’ blocks. The result might look ugly. Here is a hacky workaround for this problem:

```
adoc --texinfo | sed "s/^ / /" > outfile.texi
```

1.13 adoc.guide/Index

Index
*****

The index is under development.