
Utroff

(Beta version : user interface may change)

Utroff readme	2	Trxtr manual	51
Utroff manual	6	Dpage manual	52
U manual	9	Changes of Utroff	53
Utmac manual	11	Changes of U	54
U-ref manual	18	Changes of utmac	55
Idx manual	21	Changes of idx	58
Soelim manual	24	Changes of soelim	59
Felim manual	26	Changes of ugrind	60
Slelim manual	27	Changes of refer	62
Ugrind manual	28	Changes of tchars	64
Refer manual	31	Changes of troffxml	65
Refer input and output format	34	Changes of dpage	67
Sortbib manual	38	Two clause BSD License	68
Inv manual	40	Original BSD License	69
Mkey manual	42	Common Development and	
Hunt manual	44	Distribution License Version 1.0	70
Tchars manual	46	Tables	76
Prexml and postxml manual	49		

Utroff readme

Presentation of Utroff

Utroff is a set of macro, pre-processors, and post-processors to use easily *Heirloom Troff* today. It handles the *Heirloom troff* typographic features (paragraph at once adjustment, hanging characters, OpenType fonts, utf8), provides default support for index, table of contents, summary, bibliography, syntax highlighting, polytonic greek, and it is able to produce pdf, xml, html, fodt, man, plain text and markdown files.

Utroff is distributed in two forms : a pre-built archive with a version number for quick installs, and a development archive.

Quick start

Dependencies

Utroff depends on Heirloom-doctools¹. A recipe to install Heirloom-doctools in a convenient way for utroff can be found on the 'heirloom' subdirectory of this archive. You should look at it.

By default, **utmac** needs the OpenType Libertine font².

Trxtr, from **trofftoxml**, depends on **xsltproc**, from libxslt³.

Configuration

Configure the file *config.mk* to suit your needs.

Build and install

To build, install, and clean run :

```
| make all  
| make install  
| make clean
```

In some circumstances, you might need *utroff* tools themselves to build the manual pages. In that case, build and install executables before manual pages like this :

```
| make bin  
| make installbin  
| make man  
| make installman
```

Environment variables

The easiest way to configure *utroff* is to add its bin directory at the end of your **PATH**, and use the **U** script, which will set a temporary environment for *utroff*.

```
| U man utroff  
| U troff -mus f.tr | U dpost | ps2pdf - > f.pdf
```

¹

² <<http://linuxlibertine.org>>

<<http://heirloom.sourceforge.net/doctools.html>>

³ <<http://xmlsoft.org/XSLT/>>

I

Once done, you might want to set the **UTMAC** environment variable, which contain the absolute file name of a user defined macro. This macro can be used to source a macro which sets your locale (such as *u-en* or *u-fr*), and a macro which sets your preferred font (such as *u-libertine* and *u-apollo*).

But you can also set the environment variables manually :

- **PATH** : Must contain the directory where to find *utroff* executables.
- **TROFFMACS** : Directory containing macros files. This is an *Heirloom Troff* environment variable that **utmac** also use.
- **TROFFONTS** : Comma separated list of directory containing the fonts for **troff**. One of them must contain the default *heirloom troff* fonts, one other must contain the OpenType Libertine font. This is an *Heirloom troff* environment variable that **utmac** also use.
- **REFER** (optional) : Directory containing the bibliography list for **refer**. This is a **refer** environment variable.

Conflicts

Since *groff* and *utroff* share the same tools, the directory to install *utroff* must be chosen carefully :

- If you plan to use *groff*, or think your system need it, install *utroff* in a directory which is at the end of your **PATH** environment variable, like */opt/utroff*, which is the default.
- Otherwise, you can install *utroff* in */usr/local*.

In both case, you have to check the configuration of your **man** command. If **man** expect to find the *groff* tools, add the path to this tools in the man configuration file (usually */etc/man_db.conf*) :

```

DEFINE troff groff -mandoc
DEFINE nroff /usr/bin/nroff -mandoc
DEFINE eqn /usr/bin/eqn
DEFINE neqn /usr/bin/neqn
DEFINE tbl /usr/bin/tbl
DEFINE vgrind
DEFINE refer /usr/bin/refer
DEFINE grap
DEFINE pic /usr/bin/pic -S

```

Makefile options

The makefile offers a lot of commands. You can use them from the root directory or from a sub-directory. From the root directory, you can also use the shortcut *subdir/command*.

- **bin** : Build executable and libraries.
- **man** : Build manual pages.
- **doc** : Build pdf documentation (depends on *heirloom troff*, *utroff* executables, and *libertine* font). The pdf documentation is usually not needed since it is mostly based on the manual pages.
- **all** : shortcut to build executable, libraries and manual pages.
- **installbin** : Install binaries and libraries.
- **installman** : Install manual pages.
- **installdoc** : Install pdf documentation.
- **install** : Shortcut to install executable, libraries, and manual pages.
- **uninstallbin** : Uninstall executables and libraries.
- **uninstallman** : Uninstall manual pages.
- **uninstalldoc** : Uninstall pdf documentation.
- **uninstall** : Shortcut to uninstall executable, libraries and manual pages.
- **clean** : Clean the directory by deleting builden files.

About the pre-built archive

Utroff tools written from scratch are distributed under a two clause BSD license. The other tools are modifications of the Heirloom Tools, which are mostly distributed under the CDDL-1.0 license. Here is a summary of the licenses of the tools :

- **demo** : Demonstration files. Two clause BSD license.
- **dpage** : Postscript postprocessor which put two A4 pages on an A3 sheets of paper. Two clause BSD license.
- **heirloom** : This directory contain a makefile that will install the needed heirloom tools in the directory you choosed for utroff.
- **idx** : Build and maintain indexes. Two clause BSD license.
- **refer** : Deal with references. Executables are distributed under the CDDL-1.0 license, manuals are in four clause BSD license.
- **soelim** : Soelim replace .so by the content of the file, it is distributed under the CDDL-1.0 license. Slelim escape slashes, felim replace hardcoded fonts by strings, they are distributed under the two clause BSD license.
- **share** : shared files, such as licenses and common part of the documentation.
- **tchars** : Translate ASCII code to complex utf8 characters, such as polytonic greek. Two clause BSD license.
- **troffxml** : Export troff files to various xml formats. Two clause BSD license.
- **U** : Set environment variables to use utroff easily. Two clause BSD license.
- **ugrind** : Highlight source code. Except for the file ugrindroff.c, which is distributed under a two clause BSD license, the files are distributed under the CDDL-1.0 license.
- **utmac** : A troff macro set, the heart of *Utroff*. Two clause BSD license.

About the development archive

The development archive (**utroff-dev**) contains the *RCS* revision files, and the following tools :

- **frog** : Simple development tools to deal with versions and build changelogs. Two clause BSD license.
- **skel** : Skeleton for new tools. Two clause BSD license.
- **sophia** : Sophia maintain a database of texts, philia insert references of these texts in a document — still in development. Two clause BSD license.
- **www** : Files to build the website of utroff. Two clause BSD license, except for the *awesome* font, which is distributed under the OFL license.

The makefile of the development archive offers some usefull commands for developpers and packagers :

- **co** : Check out all files (depends on RCS).
- **web** : Build files for the utroff website (depends on heirloom troff, utroff executable and binaries). Even if they can be used as a local html documentation, web files are usually not needed.
- **installweb** : Install *utroff* website locally.
- **uninstallweb** : Uninstall web files.
- **world** : shortcut to build executable, libraries, manual pages, doc files and web files.
- **installworld** : Shortcut to install executable, libraiories, manual pages, doc files and web files.
- **uninstallworld** : Shortcut to uninstall executable, libraiories, manual pages, doc files and web files.
- **pub** : Build the needed files to share the pre-built archive (README, LICENSE, VERSION, CHANGES...).
- **tar** : Tar and install the pre-built archive and the development archive.
- **scp** : Upload the website (depends on scp).
- **mrproper** : Massively clean the directory : delete built and pre-built files (README, LICENSE), and check-in all files (depends on RCS).

Help and bugs

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

Utroff manual

Name

Utroff – Use Troff !

Description

Utroff is collection of tools to use easily *Heirloom Troff* today.

Utroff session

To use efficiently *utroff*, you should process as follow.

Configure your environment

The `U(1)`¹ script gives you an easy access to the *utroff* tools, and configure some environment variables for you. Followed by a *command*, it will set the environment variables for this onliest *command*. With no argument, it can be sourced to set the *utroff* environment, or to reset your default environment.

Think

The `idx(1)`² script will help you to produce indexes of the books you're working on.

Write

The syntax of troff documents is line oriented : each command is on a line. A troff document should not contain blank lines, nor lines beginning by a blank characters. Commands are recognized by the dot '.' at the begin at the line. A line containing only a dot is ignored by troff. Comments begin by a slash followed by an apostrophe : '\'

Troff *macros* (usually named *tmac*) define the commands one can use. On *utroff*, all the *macros* follow the same syntax, which is described on the `utmac(7)`³ manual page. Here is a simple *utmac* document :

```
.DA Author name metadata
.H1 Document title
.PP
A paragraph
.PQ
A quotation\**.
.NS
A note, between NS and NE. \** is used to insert the number
of the note.
.NE
End of the quotation.
.PP
Another paragraph.
.XT \" table of content
```

¹ See p. 9.

³ See p. 11.

² See p. 21.

|

Pre-process

Pre-processors will help you to format your documents. **soelim**(1)⁴ will source files in your current document, **tbl**(1) insert tables, **eqn**(1) write equations, **refer**(1)⁵ format references — **referformat**(7) explain its syntax —, **ugrind**(1)⁶ highlight source code, **tchars**(1)⁷ write polytonic greek. Most of them define a mini-language, and will translate this mini-language to the raw troff language.

The preprocessor **prexml**(1) is used to produce *xml* files.

Format

Troff(1) will format your documents for typesetters. **Nroff**(1) will produce plain text documents and *xml* files if you use the *ux* macro. On both of them, the **-m** option will define the *macro* you want to use.

Post-process

If you used **troff**, **dpost**(1) will produce *postscript* files, **dpage**(1)⁸ will put two A4 page on an A3 sheet of paper, **ps2pdf**(1) will produce *pdf* files.

If you used **nroff** with the **-mux** option, **postxml**(1) will produce *xml* files, and **trxtr**(1) will translate the *utroff xml* format to famous *xml* formats, such as *xhtml* and *fodt*.

Recover your environment

If you sourced the **U** script, source it again to recover your default environment.

Example

All in one, a complex *utroff* session looks like :

```
| . U
| vim f.tr
| soelim f.tr | tbl | eqn | refer -p biblio.ref | ugrind \
| | tchars > f.t
| troff -mus f.t | dpost | dpage | ps2pdf - > f.pdf
| prexml f.t | nroff -mux | postxml | trxtr html > f.html
| prexml f.t | nroff -mux | postxml | trxtr fodt > f.fodt
| nroff -mut f.t > f.txt
| . U
```

But most often, you only have to do :

```
| U troff -mus f.tr | U dpost | ps2pdf - > f.pdf
```

Environment

If you don't use the **U**(1)⁹ script, you should define the following environment variables :

PATH The *utroff* executables must be the first found on the **PATH**.

MANPATH

The *utroff* manual pages must be the first found on the **MANPATH**.

TROFFONTS

Comma separated list of directory containing the fonts for troff. One of them must con-

⁴ See p. 24.

⁷ See p. 46.

⁵ See p. 31.

⁸ See p. 52.

⁶ See p. 28.

⁹ See p. 9.

tain the default heirloom troff fonts, one other must contain the OpenType Libertine font.

TROFFMACS

Directory containing the *utmac* macros files.

UTMAC

A user defined macro that utmac macros will load. This macro can be used to source a macro which sets your locale (such as *u-en* or *u-fr*), and a macro which sets your preferred font (such as *u-libertine* and *u-apolline*).

See also

Utroff manual pages : **U**(1)¹⁰, **idx**(1)¹¹, **utmac**(7)¹², **u-ref**(7). **soelim**(1)¹³, **felim**(1), **slelim**(1), **ugrind**(1)¹⁴, **tchars**(1)¹⁵. **refer**(1)¹⁶, **referformat**(7), **sortbib**(1), **hunt**(1), **inv**(1), **mkey**(1). **dpost**(1), **dpage**(1)¹⁷, **prexml**(1), **trxtr**(1).

Some Heirloom Troff manual pages : **otfdump**(1), **nroff**(1), **troff**(1), **dpost**(1). **tbl**(1), **eqn**(1), **eqnchar**(7), **checknr**(1), **pickpack**(1), **grap**(1), **pic**(1).

Other usefull manual pages : **ps2pdf**(1).

License

This man page is distributed under a two clause BSD¹⁸ license.

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Author

Written by Pierre-Jean FICHET.

¹⁰ See p. 9.

¹¹ See p. 21.

¹² See p. 11.

¹³ See p. 24.

¹⁴ See p. 28.

¹⁵ See p. 46.

¹⁶ See p. 31.

¹⁷ See p. 52.

¹⁸ See p. 68.

U manual

Name

U – Set environment variables to use *Utroff*

Synopsis

@BINDIR@/U [**command**]

. @BINDIR@/U

Description

Utroff, *Heirloom Troff* and *Groff* share some of the same utilities, even if their source code differ. To avoid conflicts, it is wise to put the *utroff* utilities at the end of your **PATH**, which is not hand-full for a daily usage. The **U** script will set your **PATH** and some environment variable to use *utroff* easily.

Followed by a *command*, **U** will set the environment variables for this onliest *command*. With no argument, it can be sourced to set or reset the *Utroff* environment.

The following environments variable are currently set by **U** :

PATH To be sure that the *utroff* executables are the first found on the **PATH**.

MANPATH

To be sure that the *utroff* manual pages are the first found on the **MANPATH**.

TROFFONTS

To find the Libertine OpenType font.

TROFFMACS

To find the *utmac* macros files.

PS1 When sourced, **U** lightly modify your prompt.

There is one important environment variable that **U** does **not** set :

UTMAC

A user defined macro that *utmac* macros will load. This macro can be used to source a macro which sets your locale (such as *u-en* or *u-fr*), and a macro which sets your preferred font (such as *u-libertine* and *u-apolline*).

Example

```
# command
U troff -muh f.tr | U dpost | ps2pdf -> file.pdf
U ugrind g.tr | U nroff -mut > g.txt
U man refer
# session
. U
troff -muh f.tr | dpost | ps2pdf - > file.pdf
ugrind g.tr | nroff -mut > g.txt
man refer
. U
```

|

See also

utroff(7).

License

U and this man page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

Author

Written by Pierre-Jean FICHET.

Utmac manual

Name

utmac : macro set for heirloom troff.

Description

The *U troff macros* — *utmac* — is a set of *troff* macros which uses the typographic features of *Heirloom Troff*. Bibliography, indexes, table of content and summaries can be automatically inserted. *Utmac* informs about orphans, widows and blanks at the bottom of page, and provides macros to help the user to fix them manually. *Utmac* can produce terminal output, high quality pdf files, ReStructuredText files, plain text file, and clean xml files — and so, with some xsl stylesheet, flat open document texts (fodt) and html files.

Files

User macro files

The following macro files can be used by the user (called with the **-m** option of troff) :

uh Made with humanities in mind. It use marginal notes instead of footnotes.
um Translate a *utmac* document to manpage with *nroff*.
ut Translate an *utmac* document to a flat text file (such as README files) with *nroff*.
us Made for technical documentation. It use a common layout, almost like the old troff *ms*.
uw Translate an *utmac* document to the markdown wiki format.
ux Translate an *utmac* document to xml. This macro must be used with the `postxml(1)` post-processor.

Locale macro files

Utmac use the **UTMAC** environment variable to load a locale macro. The file `@MACDIR@/u-locale` is an example of such a macro, you should copy and edit it to suit your needs.

Internal macro files

The following macros files are internally used by *utmac* :

u-ref Format a refer bibliography. See `u-ref(7)`.
u-grind Handle *ugrind* pre-processor.
u-idx Inline and index macros.
u-en u-fr
 Localization macros, for english and french. They should be loaded by the file defined by the **UTMAC** environment variable. If the **UTMAC** environment variable is not defined, english is loaded.
u-apolline, u-libertine
 Font macro, for the opentype apolline and libertine font. If the **UTMAC** environment variable is not defined, *u-libertine* is loaded.

Macros

An *utmac* document should be structured in four parts : metadatas, definition of some register, content, and appendix. The content can contain header macros, summary macros, paragraph macros, typographic macros, note macros, inline macros, and *utmac tests* macro.

All macros have two letters ; the first one define the gender of the macro (metaDAtas, Register, Utmac tests, Heading, Summary, Paragraph, Typographic, Notes, inLine, AppendiX).

Metadata macros

These macro are used to populate the metadatas of the final file (pdf, xml, or html). They must be inserted in the first page of the document – their natural place being the first lines of the file.

DA *author name*

Define author metadata.

DI *Identifier*

Define identifier (RCS Id..).

DK *list of keywords*

Define keywords metadata.

DS *subject of the text*

Define subject metadata.

DT *title of the text*

Define title metadata.

Register macro to define format and layout

Some number register are used by *utmac* to make or not some actions. Set to 1, the action is done, set to 0, the action is not done. The following macro can be used to define the state of this number register. The name of the macro is also the name of the number register. For example, **.RV 1** acts like **.nr RV 1**. This is usefull to format a document from the command line :

```
| troff -rRV=1 -muh file
```

RV *num*

If *num*=1, use a recto-verso layout. Default is 0.

RI *num* If *num*=1, replace references by *Idem* when needed. Default is 1.

RH *num*

If *num*=1, print help message. Help message inform about orphans, widows, blank at the bottom of pages, note diverted to next page, and multiple definition of anchor keyword. Default is 1.

RL [*fr|en*]

Set language of file. Use this macro to localize some string and to respect local typographic standard if you want to override the default which should be defined by the macro in the **UTMAC** environment variable.

RO *num*

If *num*=1, replace some part of a reference (issuer, city and date) by *op. cit.* if the same reference has appeared in the text. Default is 1.

RP *num*

If *num*=1, print page number. Default is 1.

RN *num*

If *num*=1, print reference in a note, if *num*=0, the reference is printed directly in the text, by using the **PQ** macro. Default is 1.

Utmac tests macro

To adapt the content of the document to the macro which is used, *utmac* provide a simple test macro :

US [*!*] *utmac-macro command*

Execute *command* if *utmac-macro* is used to format the document ; or, if a *!* is the first argument, execute *command* if *utmac-macro* is *not* used to format the document. If *command* is preceded by a "." or a "'", it'll be executed, else, the text of *command* is simply written in the document. **US** macros can be nested to make complex tests.

The following macro are testable : *uh, um, ut, us, uw, ux, u-ref, u-grind, u-idx, u-en, u-fr, u-apolline, u-libertine, u-locale.*

For example, here is how to write a special heading for the um and ux macros :

```
.US um .H2 UTMAC 7 \" Manual page heading
.US ux .H2 UTMAC(7) \" Xml heading
.US ! um .US ! ux .H2 Utmac manual page \" Default heading
```

Heading macros

These macro are used to print headings and to record them for a summary or a table of contents.

H1 *Title of the book*

Book heading.

H2 *Title of the chapter*

Chapter heading.

H3 *Title of the paragraph*

Paragraph heading.

H4 *Subheading*

Subheading.

H0 *Temporary title*

Temporary heading, printed in the header or in the footer of the current page. This heading is not recorded in the table of content.

H* *Margin title*

Heading printed in header or footer of the following pages, and in the current ones if H0 is not set for this page. This heading is not recorded in the table of content.

Summaries macros

To print a summary, two or three pass of troff are needed : the first one to record the summary, the second one to print it, and a third one might be necessary to adjust the page number if the summary is longer than one page.

S2 Print a summary of chapter headings. The summary is printed on a new page.

S3 Print a summary of paragraph headings. The summary is printed on the actual page.

S4 Print a summary of subheading. The summary is printed on the actual page.

S* Print the whole table of content as a summary.

Paragraph macros

PB Left indent (extended) paragraph. Usefull to begin some sort of lists.

PC Centered paragraph.

PP Justified paragraph.

PQ Quote paragraph.

PI [*bullet*]

List paragraph. Default bullet is usually an emdash (*(\em)*).

PL Left align paragraph.

PR Right align paragraph.

PX Example paragraph. Text is not filled.

Typographic macros

TD Use micro-typography to decrease the width of a paragraph. It can be inserted at the beginning (just after *PP*) of a well chosen paragraph to avoid orphans or widows. A well chosen paragraph is a long one, with a short last line.

TI Use micro-typography to increase the width of a paragraph. It can be inserted at the beginning (just after *PP*) of a well chosen paragraph to avoid a blank at the bottom of a page. A well chosen paragraph is a long one, with a long last line.

Note macros

NS *no* Start a numbered note. The default is to print a number before the note, but if an argument is given, no number will be printed. To insert a number reference to the note in the text, use the * string, say insert : *(**)* where you want the number to appear.

NT Print a note on top of the others, or in the flow of the text, depending of the macro

- used.
- NB** Print a note in the bottom of the page, or in the flow of the text, depending of the macro used.
- NE** End *NS*, *NT* or *NB*.

Inline macros

Inline macros add semantic information to the source and the final document, define links and anchors, and — for some of them — index words.

To deal with all these situations, inline macro have two usage :

- By default, an inline macro does an invisible job and doesn't print anything ;
- But, if the inline macro is preceded by the **LP** macro, it will do its invisible job and print the arguments of **LP** macro.

The following example first index a name (**LN** macro), and next print (**LP**) a link to an url (**LU**) address :

```

The
.LN Pierre-Jean Fichet
author of \*Iutmac\*R share them on a
.LP website .
.LU http://utroff.org

```

The following inline macro are defined :

- LA** *acronym meaning*
Define an acronym and its meaning, and index them.
- LC** *command*
Define a command (does nothing).
- LD** *date* Define a date (does nothing).
- LE** *error*
Define an error (does nothing).
- LK** *keyword*
Insert an anchor named *keyword*.
- LL** *keyword*
Insert a link to the anchor *keyword*. The text to link is defined by the **LP** macro.
- LM** *email*
Insert a link to an email address. The text to link is defined by the **LP** macro.
- LN** Define a name and index it.
- LO** Define a name of thing (object) and index it.
- LP** Define text to be printed by an inline macro. The first argument is the text itself, the second is written after the text, and the third one is written before it.
- LT** *title* Define a title, and index it.
- LU** *url* Insert a link to an url. The text to link is defined by the **LP** macro.
- LW** Define a word and index it.

Appendix macros

- XB** [*bibliography*]
Print a bibliography. If *bibliography* is omitted, mh will use the *\$REFER* environment variable. The bibliography begin with the **H3** macro. *Utmac* try to sort the bibliography using *refer -d* option, but this doesn't give correct result on some bibliography list. For a perfect result, you should sort your bibliography with *sortbib*. See **u-ref(7)**.
- XI** [*list of index to print*]
Print all the indexes listed as argument, if existing. If no argument is given, print all the existing indexes. To index something, use an inline macro (the second letter of the inline macro is the index name). Each index begin with the **H3** macro.
- XR** Print an index of the title which appeared in references. Is equivalent to **.XI T**.

XT Print a table of content. The table begin with the **H3** macro.

Fonts

To access a font with *utmac*, you *must* use a string instead of the troff `\f` and `.fp` commands. This is needed to avoid to know which font will use *utmac* at which moment : the string has a different meaning depending of the context. So, to insert a single letter font, you should write `*I`, a two letter font `*(BI`, and a more than two letter font `*[BIC]`. Strings are defined as follow :

R	Roman font.
B	Bold font.
I	Italic font.
BI	Bold-Italic font.
C	Small Capital font.
BC	Bold Small Capital font.
IC	Italic Small Capital font.
BIC	Bold Italic Small Capital font.
A	Acronym font (capital to small capital).
BA	Bold Acronym font.
IA	Italic Acronym font.
BIA	Bold Italic Acronym font.
S	Superscript font.
BS	Bold Superscript font.
IS	Italic Superscript font.
BIS	Bold Italic Superscript font.
F	Final font (glyphs that might only appear at the end of a line).
BF	Bold Final font.
IF	Italic Final font.
BIF	Bold Italic Final font.
P	Return to previous font.

Naming conventions

Utmac use some convention to name macro, strings, and number register. Internally, files are divided in modules, and the naming conventions are as follow :

- The name of macros are of the form **module :name**.
- The name of strings and number registers are of the forme **module-name**.

Externally, all macro only contain uppercase letter and is never longer than three letters.

So, *utmac* reserve the following names :

- Names containing the characters `:` and `-`.
- Names containing uppercase letters, and shorter than four letters.

Example

The following example is a fully commented simple *utmac* document. As a summary is inserted, two pass of *troff* are needed. If the summary is longer than one page, a third pass of troff would be needed to handle correctly the page references, but this is not the case in this short example. So, this document can be build using these commands :

```
| troff -muh alice.tr > /dev/null
| troff -muh alice.tr | dpost | ps2pdf - alice.pdf

| .\ " Start example
| .\ " Use recto verso
| .RV
```

```

.\" Define metadatas
.DA Lewis Carroll
.DT Alice's adventures in wonderland and Through the looking glass
.DK alice nabuchodonosor jabberwocky
.\" Build first page
.H0 Lewis Carroll
.H1 Alice's adventures
.\" Insert a summary of chapter headings
.S2
.\" Chapter heading
.H2 Alice's adventures in wonderland
.\" A short exergue
.NT
All in the golden afternoon
Full leisurly we glide
.NE
.\" First paragraph
.H3 Down the rabbitt hole
.\" Some text with font definitions, and a note appeal
.PP
\*CAlice\*P was beginning to be very tired of sitting by her
sister on the bank, and of having nothing to do: once or
twice she had peeped into the book her sister was reading,
but it had no pictures or conversations in it, "\*Iand what
is the use of a book\*P" thought \*CAlice\*P, "\*Iwithout
pictures or conversations?\*P\*"
.\" A note
.NS
Yes, what is the use of such a book?
.NE
.\" Some other chapters and paragraphs...
.H3 The pool of tears
.H2 Through the looking glass
.\" Table of content
.XT
.\" Stop example

```

Environment

TROFFMACS

This is a *troff* environment variable, that *utmac* use to find tmacs file. It must be defined for *utmac* to work.

TROFFONTS

This is a *troff* environment variable that *utmac* use to find font files. By default, this variable should contain the path to the opentype libertine font.

UTMAC

Locale tmac file that *utmac* will load. It should contain localization macros. See `@MACDIR@/u-locale` for an example.

Files

The following files are installed on the directory `@MACDIR@ : @BIN@`

See also

u-ref(7), postxml(1), idx(1), refer(1).

License

Utmac macros and this manual page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

Author

Pierre-Jean FICHET.

U-ref manual

Name

U-ref : Format references for the U troff macros.

Description

The *u-ref* macro manage references in *refer* format for the *U troff macros* set (*utmac*). *u-ref* has high level built-in capabilities :

- It formats the bibliography following the *iso-690* bibliography standart which define a lot of fields, and which fix the order on which they must appear.
- It formats differently footnotes references and bibliography list to increase readability.
- It replaces yet cited references by "*Idem.*" or "*op. cit. hereinbefore*" or "*op. cit. p xx.*", depending if the references has been cited just before, in the same page, or in the page *xx* of the document. If only the heading (**%h** field) or or the page (**%p** field) of the reference differ, they are simply written after these replacement strings.
- It builds an index of references that can be printed with the **.AR** macro.

u-ref use a new version of *refer* instead of the original *heirloom refer* to handle correctly the **ISO-690** bibliography standart.

Usage

Macros

The following macro can be used by the user :

AB *file.ref*

Print a bibliography list of the file *file.ref*.

AR Print an index of references.

RI Reverse usage of *Idem* in reference. The first call of this macro will remove the usage of *Idem*, so the reference will never be replaced by "*idem*". Default is to replace all informations similar to the previous reference.

RO Reverse usage of *op. cit.* in reference. The first call of this macro will remove the usage of *op. cit.*, so the reference will never be replaced by "*op. cit.*". Default is to replace issuer, city and date by *op. cit. p. xx* if the same reference appears before in the text.

RN Reverse usage of notes in reference. The first call of this macro will remove the usage of the note to print references, but use the **PQ** macro to print them in the text. Default is to enclose all references with **NS** and **NE** to print them as a notes.

Example

- Use the **.[** and **.]** *refer* macro to insert the reference. The first line following the **.[** macro must contain words to find the correct reference. Following lines can contain fields definition.
- Since *u-ref* write the reference in a note by default, you must add the ****** string to insert, where appropriate, a note number in the text.
- Since the note number is inserted manually, you should use the bare mode of *refer* (the **-b** option), so that the *refer* command line is on the form : **refer -b -p file.ref**.

| He said: «a short citation**».

| **.[**

```

Keywords...
%p 3
.]

```

Fields

u-ref use uppercase and lowercase fields. The conventional meaning of each field is as follows :

- A** The name of an author. If the name contains a title such as *Jr.* at the end, it should be separated from the last name by a comma. There can be multiple occurrences of the **A** field. The order is significant. It is a good idea always to supply an **A** field or a **Q** field.
 - B** For an article that is part of a book, the title of the book.
 - C** The place (city) of publication.
 - D** The date of publication. The year should be specified in full. If the month is specified, the name rather than the number of the month should be used. It is a good idea always to supply a **D** field ; if the date is unknown, a value such as *in press* or *unknown* can be used.
 - E** For an article that is part of a book, the name of an editor of the book. For a serie of journals, the responsible of the serie. For a serie of books, the responsible of the serie. Refer man page say that where the work has editors and no authors, the names of the editors should be given as **A** fields and (*ed*) or (*eds*) should be appended to the last author. On the one hand this might be a good idea for sorting purpose and to be sure that the name of the editor appear before the title of the book, but on the other end, the standart says that *ed* should be written without parenthesis, thing that might be confusing for *refer*. You should try both solution and choose the best one in each situation.
 - F** Internally used by *refer* : the number of the reference. It must should be defined by user.
 - G** For a book, the ISBN (International Standard Book Number). For a Journal, the ISSN (International Standard Serial Number).
 - H** Information about the publication : first edition, french edition, etc.
 - h** Heading of the section of the book where the citation appears.
 - I** The publisher (issuer).
 - J** Defines a periodical publication : For an article in a journal, the name of the journal. For a journal from a serie, the name of the journal. For a serie of journals, the name of the serie.
 - K** Keywords used for searching. This field is required by *u-ref* to replace citations by *Idem.* and co. It must be defined for *u-ref* to work properly, and should contain only ascii characters.
 - L** Label. Not used by *u-ref*.
 - M** The collection in which the reference is published.
 - N** Journal issue number.
 - O** Other information. This is usually printed at the end of the reference, before the field G. It's the place to mention the support of an electronic document (e-mail, pdf, html...)
 - P** **Total** number of page of the book.
 - p** Exact page of the citation, or position of the article in a book. A range of pages can be specified as **m-n**.
 - Q** The name of the author, if the author is not a person. This will only be used if there are no **A** fields. There can only be one **Q** field.
 - R** Technical or scientific report number.
 - S** Series name.
 - T** Title. For an article in a book or a journal, this should be the title of the article.
 - U** Co-author (translator, illustrator...). Should be preceded by its function : *tr.* for a translator, etc.
 - V** For a book in a serie, the volume number of that book. For a whole serie, the number of books in that serie.
 - w** Type (**which**) of an electronic document (online, CD-ROM...)
 - X** Annotation.
 - x** URL of an online document. *u-ref* doesn't have macro to hyphenate correctly urls. In
-

you bibliography database, you should use the special character \ : to tell troff where to hyphenate the url, for example like this :

- | %x http://\/:example.\:com/
- y Update date of an electronic document.
- z Date of last access to an electronic document.

For all fields except **A** and **E**, if there is more than one occurrence of a particular field in a record, only the last such field will be used.

Files

@MACDIR@/u-ref, the macro file.

See also

iso-690, utmac(7), refer(1), referformat(7)

License

u-ref macro and this manual page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Author

Pierre-Jean Fichet.

Idx manual

Name

Idx – make and manipulate index files.

Synopsis

@BINDIR@/idx [-dhmptw] *file*

Description

Idx is an helper tool to make and to manipulate index files. *Idx* has been made to automatically index troff document, but it can also be used manually, to make an index of a book.

Options

- d Take troff output in entry, and output an index in a troff format. Internally used by troff macros to build indexes. This is the default.
- h Print a short help.
- m Mix all lines in a single line.
- p Take a list of words in entry, and output a list of pages.
- r Reformat a list of words to fix errors.
- t Take a list of pages in entry, and output an index in a troff format.
- w Take a list of pages in entry, and output a list of words.

Format

There's four format of index file that *Idx* can work with : a list of pages, a list of words, troff output and troff input.

List of pages

A list of pages indicates which words appears on a page. Each line of a list of pages is formatted as follow :

- page number,
- two semi-colons followed by one space (: :),
- a list of words separated by a comma and a space (,).

Example :

```
33:: troff, refer, index
34:: troff, index, nroff
35:: index, unix, refer
```

List of words

A list of words indicates in which pages appears a word. Each line of a list of words is formatted as follow :

- the word,
 - two semi-colons followed by one space (: :),
 - a list of page number or range of pages separated by a comma and a space (,). A range
-

of page is made by two page number separated by a tirect (-).

The previous example would be formatted as follow :

```
index:: 33-35
nroff:: 34
refer:: 33, 35
troff:: 33
unix:: 35
```

Troff output

When *idx* is used to index a troff document, the output produced by troff and passed to *idx* should be as follow :

- A single letter key which indicate what the index is about (W for index of words, T for index of titles...),
- A ">" followed by a space (>),
- The indexed word,
- Two semicolons followed by a space (: :),
- The page number where this word appears.

The previous example should be formatted as follow :

```
W> troff:: 33
W> refer:: 33
W> index:: 33
W> troff:: 34
W> index:: 33
W> nroff:: 34
W> index:: 35
W> unix:: 35
W> refer:: 35
```

Troff input

Idx can format an index so that troff can print it correctly. The format of troff input is as follow :

```
.K<           \ " one letter keyword defining the index
.ds <P x, m-n \ " list of pages
.K> word      \ " the word which appears on these pages
```

So, a troff macro that use *idx* should define the following macros :

- K<** Start an index of type K. Usually, such a macro will print a title, such as "Index of words".
- K>** Print the index entry defined as argument, followed by the list of pages previously defined by the <P string. You should add a dot at the end of the list of pages, since *idx* will not add it.

Example

This example taken from real life will show how to print an index of a paper book.

- Read the book, and, annotate it to underline the words you'll have to index.
- On a file of your computer, write an index formatted as a list of pages : for each page of the book, write the words it contains.
- Use *idx* to translate this file to a sorted list of words :

```
| idx -w a.idx > b.idx
```

- You should read this list of words, since you'll probably have to correct some mistakes,

-
- or rename some index entry.
 - Once done, reformat it with `idx` :

 - ▮ `idx -r b.idx > c.idx`
 - Use `idx` to pass this formatted index to `troff` :

 - ▮ `idx -t c.idx | troff | dpost > index.ps`

Compatibility

Idx is needed by `utmac(7)`¹ to build indexes. In that case, if heirloom *troff* is used, and if heirloom tools are in the beginning of the path, be sure that `'awk'` links to heirloom `'nawk'` and not heirloom `'oawk'`.

Files

`@BINDIR@/idx`

See also

`utmac(7)`².

License

Idx and this man page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at `help` at `utroff dot org`. Please, send bugs and patches at `help` at `utroff dot org`.

Author

Written by Pierre-Jean FICHET.

¹ See p. 11.

² See p. 11.

Soelim manual

Name

soelim – (utroff) eliminate .so's from nroff input

Synopsis

@BINDIR@/soelim [*file*] ...

Description

Soelim reads the specified files or the standard input and performs the textual inclusion implied by the *nroff* directives of the form

```
.so somefile
```

when they appear at the beginning of input lines. This is useful since programs such as *tbl* do not normally do this; it allows the placement of individual tables in separate files to be run as a part of a large document.

Note that inclusion can be suppressed by using ‘’ instead of ‘.’, i.e.

```
| 'so /usr/ucblib/doctools/tmac/s
```

This version of *soelim* keep trace of real the file name and line number, so that troff error messages still gets meaningfull.

A sample usage of *soelim* would be

```
| soelim exum?.n | tbl | nroff -ms | col | lp
```

See also

more(1) , felim(1), slelim(1).

Notes

The format of the source commands must involve no strangeness – exactly one blank must precede and no blanks follow the file name.

License

Soelim is distributed under the CDDL-1.0 license and this man page are distributed under the original BSD license.

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Authors

Written by William JOY. Modified by Pierre-Jean FICHET.

Felim manual

Name

felim – (utroff) eliminate hardcoded fonts from nroff input

Synopsis

@BINDIR@/felim < file

Description

felim read the standard input and replace hardcoded troff font definitions (`\fx`) by strings (`*x`). This is usefull since some troff macro expect such string defined fonts.

See also

soelim(1), slelim(1).

License

felim and this man page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Author

Written by Pierre-Jean FICHET.

Slelim manual

Name

slelim – escape some slash from nroff input

Synopsis

@BINDIR@/slelim < file

Description

When transferring a troff document to another format of troff document using nroff, the slash written in the sources usually means that the string following should be simply written as is. *Slelim* try to handle these situation.

slelim read the standard input and replace or not the slash character following these rules :

- Inside **.vS** and **.vE** macros, nothing is done.
- If the slash is followed by a newline, an **'*** an **'n'** or another slash, *slelim* does nothing.
- If the slash is followed by an **'f'**, the **'f'** is replaced by an **'**'**, so that font definitions are replaced by a string definition.
- in all other case, the slash is escaped, ie duplicated.

See also

soelim(1), felim(1).

License

slelim and this man page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Author

Written by Pierre-Jean FICHET.

Ugrind manual

Name

ugrind – troff preprocessor which highlight source code

Synopsis

@BINDIR@/ugrind [-**d** *file*] [-**h** *header*] [-**l** *language*] [-**t** *size*] *name* ...

Description

Ugrind insert string definition in files to highlight the source code they contain. *Ugrind* runs in two basic modes : regular mode or whole file mode.

- In regular mode, *ugrind* only highlight code which is between the **vs** and **vE** macros. The language is specified by the first argument of the **vs** macro.
- In whole file mode, *ugrind* highlight the whole file. The language is specified by the **-l** argument.

The options are :

- d** Specifies an alternate language definitions file (default is @LIBDIR@/ugrinddefs).
- h** Specify a header string. The header will be printed as argument of the **.vH** macro.
- l** Enter whole file mode, and pecifies the language to use.
- t** Specify a nondefault tab size (default is 4).

Currently, known languages are C (**-lc**), C++ (**-lc++**), CSH (**-lcsh**), FORTRAN (**-lfortran**), ICON (**-li**), ISP (**-lisp**), LDL (**-ldl**), LISP LIKE (**-lm**), MODEL (**-lm**), MODULA2 (**-lmod2**), PASCAL (**-lp**), POSTXML (**-lpostxml**), RATFOR (**-lr**), RUSSEL (**lrussell**), SHELL (**-lsh**), TROFF (**-ltr**), XML (**-lxml**), and YACC (**-lyacc**),

Programming style

Ugrind assumes that a certain programming style is followed :

C and C++

Functions names can be preceded on a line only by spaces, tabs, or an asterisk. The parenthesized arguments must also be on the same line.

PASCAL

Function names need to appear on the same line as the keywords *fuctions* or *procedure*.

MODEL

Function names need to appear on the same line as the keywords *is beginproc*.

TROFF

Between **.vs** and **.vE** macros in regular mode, and in the whole file in whole file mode, *ugrind* will take care to escape all the commands, macros, strings and number register that could be interpreted by troff. That means that to highlight troff source code, one juste has to write the code as is. That also means that it is not possible to insert formatting commands inside the code that has to be highlighted.

More generally, arbitrary formatting styles for programs mostly look bad. The use of spaces to align source code fails miserably ; if you plan to *ugrind* your program you should use tabs. This is somewhat inevitable since the font used by *ugrind* is variable width.

Macros and strings

To highlight code, *ugrind* insert strings in the file, and use some macros, so, to format the output of *ugrind* with *troff*, the following strings and macros must be defined by your macro file.

Macros

- vS In regular mode, start a code block to highlight. The first argument define the language. See the `-l` option to get the list of supported languages.
- vE In regular mode, end a code block to highlight.
- vF In whole file mode, start a new file. The first argument is the name of the file.
- vH The header defined by the `-h` option is defined as argument of this macro.
- vO Indicate that a function begins on the next line. The first argument is the function name.
- vC Indicate that a function gets closed on the previous line.

Strings

- +C Begin a comment,
- C End a comment,
- +K Begin a keyword,
- K End a keyword,
- +S Begin a string,
- S End a string,
- +V Begin a variable,
- V End a variable,
- e a slash (\),
- a tirect (-),
- _ a bottom tirect (_).

History

Ugrind is a hack of *vfontedpr* (that probably means *visual-font-edit-print*), which is the preprocessor used by the mostly forgotten *vgrind* script, which aims to visually "grind nice listing of programs" with *troff*. The code that *vfontedpr* use to read the language description file is "stolen almost totally from *termcap*". *Vgrind* and *vfontedpr* have been written by Dave Presotto and William Joy, for 4.3BSD-Tahoe. They have been modified by Gunnar Ritter as part of the *Heirloom documentation tools*.

License

Ugrind and its man page are distributed under the original BSD license. The code which highlight *troff* language, written from scratch, is distributed under a two clause BSD license. The language description file (`@LIBDIR@/ugrinddefs`) is distributed under the CDDL 1.0 License.

Files

- `@BINDIR@/ugrind` preprocessor
- `@LIBDIR@/ugrinddefs` language descriptions

See also

- `troff(1)`

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Authors

Written by Dave PRESOTTO and William Joy. Modified by Gunnar RITTER and Pierre-Jean Fichet.

Refer manual

Name

refer – (utroff) insert iso-690 compliant references in document

Synopsis

@BINDIR@/refer [-abcdeinPS] [-fn] [-kx] [-lm,n] [-p bib] [-skeys] [-Bl.m] [file ...]

Description

Refer is an enhancement of the *Heirloom refer* preprocessor for *nroff* and *troff* that finds and formats references for footnotes or endnotes. It is also the base for a series of programs designed to index, search, sort, and print stand-alone bibliographies, or other data entered in the appropriate form.

This version adds to *Heirloom refer* the ability to sort following the *iso-690(7)* bibliography standard (-i option), a better capitalize option (-d), support editor names like authors names, and insertion of the universal "&" instead of the english "and" to join names.

Given an incomplete citation with sufficiently precise keywords, *refer* will search a bibliographic database for references containing these keywords anywhere in the title, author, journal, etc. The input file (or standard input) is copied to standard output, except for lines between .[and .] delimiters, which are assumed to contain keywords, and are replaced by information from the bibliographic database. The user may also search different databases, override particular fields, or add new fields. The reference data, from whatever source, are assigned to a set of *troff* strings. Macro packages such as *ms* print the finished reference text from these strings. By default references are flagged by footnote numbers.

The following options are available :

- a *n* Reverse the first *n* author or editor names (Jones, J. A. instead of J. A. Jones). If *n* is omitted all author and editor names are reversed.
- b Bare mode : do not put any flags in text (neither numbers nor labels).
- c *keys* Capitalize (with CAPS SMALL CAPS) the whole field whose key-letters are in the first-name part of fields whose key-letters are in *keys*. The firstname part is the last word of the fields or the previous one if it ends with a comma (",").
- d *keys* Insert strings around the lastname, firstname and junior part of fields whose key-letters are in *keys*, for later formatting. The firstname part is the last word of the field or the previous one if it ends with a comma (","). Strings are *(+F and *(-F around firstname, *(+L and *(-L around lastname, *(+J and *(-J around junior.
- e Instead of leaving the references where encountered, accumulate them until a sequence of the form

```
| .[
| $LIST$
| .]
```

is encountered, and then write out all references collected so far. Collapse references to same source.

- f *n* Set the footnote number to *n* instead of the default of 1 (one). With labels rather than
-

-
- numbers, this flag is a no-op.
 - i Sort following the iso-690 standard order of fields : If **J** is defined, the order of fields is "**QATESVBJ**"; Else, if **B** is defined, the order of fields is "**QATESVB**"; Else, the order of fields is "**QAESVT**". The *u-ref* macro need this option to work properly.
 - k *x* Instead of numbering references, use labels as specified in a reference data line beginning %*x*; by default *x* is **L**.
 - l *m* , *n* Instead of numbering references, use labels made from the senior author's last name and the year of publication. Only the first *m* letters of the last name and the last *n* digits of the date are used. If either *m* or *n* is omitted the entire name or date respectively is used.
 - n Do not search the default file `/$REFLIB$/papers/Ind`. If there is a **REFER** environment variable, the specified file will be searched instead of the default file ; in this case the **-n** flag has no effect.
 - p *bib* Take the next argument *bib* as a file of references to be searched. The default file is searched last.
 - s *keys* Sort references by fields whose key-letters are in the *keys* string ; permute reference numbers in text accordingly. Implies **-e**. The key-letters in *keys* may be followed by a number to indicate how many such fields are used, with + taken as a very large number. The default is **AD** which sorts on the senior author and then date ; to sort, for example, on all authors and then title, use **-sA+T**.
 - B *l.m* Bibliography mode. Take a file composed of records separated by blank lines, and turn them into *troff* input. Label *l* will be turned into the macro *.m* with *l* defaulting to %**X** and *.m* defaulting to **.AP** (annotation paragraph).
 - P Place punctuation marks ., : ; ? ! after the reference signal, rather than before. (Periods and commas used to be done with strings.)
 - S Produce references in the Natural or Social Science format.

To use your own references, put them in the format described below. They can be searched more rapidly by running *indxbib*(1B) on them before using *refer*; failure to index results in a linear search. When *refer* is used with the *eqn*, *neqn* or *tbl* preprocessors *refer* should be first, to minimize the volume of data passed through pipes.

The *refer* preprocessor and associated programs expect input from a file of references composed of records separated by blank lines. A record is a set of lines (fields), each containing one kind of information. Fields start on a line beginning with a "%", followed by a key-letter, then a blank, and finally the contents of the field, and continue until the next line starting with "%". The output ordering and formatting of fields is controlled by the macros specified for *nroff/troff* (for footnotes and endnotes) or *roffbib* (for stand-alone bibliographies). For a list of the most common key-letters and their corresponding fields, see *addbib*(1B). An example of a *refer* entry is given below.

Example

%A	M. E. Lesk
%T	Some Applications of Inverted Indexes on the UNIX System
%B	UNIX Programmer's Manual
%V	2b
%I	Bell Laboratories
%C	Murray Hill, NJ
%D	1978

Files

@REFDIR@/papers directory of default publication lists @REFDIR@ directory of companion programs

See also

referformat(7), sortbib(1), mkey(1), inv(1), hunt(1), and *Some application of Inverted Indexes in the UNIX System* by M. E. LESK.

Notes

Blank spaces at the end of lines in bibliography fields will cause the records to sort and reverse incorrectly. Sorting large numbers of references causes a core dump.

License

Refer is distributed under the CDDL-1.0 license. This manual page is distributed under the original BSD license.

Help and bugs

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

Authors

Written by Mike LESK, modified by Pierre-Jean FICHET.

Refer input and output format

Refer file

A reference file is a set of bibliographic references usable with *refer*. It can be indexed using the softwares *inv* and *mkey* for fast searching.

The strings defined by *refer* are taken directly from the files of references, which are in the following format. The references should be separated by blank lines. Each reference is a sequence of lines beginning with % and followed by a key-letter. The remainder of that line, and successive lines until the next line beginning with %, contain the information specified by the key-letter. In general, *refer* does not interpret the information, but merely presents it to the macro package for final formatting. A user with a separate macro package, for example, can add new key-letters or use the existing ones for other purposes without bothering *refer*.

The meaning of the key-letters given below, in particular, is that assigned by the *-ms* macros. Not all information, obviously, is used with each citation. For example, if a document is both an internal memorandum and a journal article, the macros ignore the memorandum version and cite only the journal article. Some kinds of information are not used at all in printing the reference; if a user does not like finding references by specifying title or author keywords, and prefers to add specific keywords to the citation, a field is available which is searched but not printed (K).

The key letters currently recognized by *refer* and *-ms*, with the kind of information implied, are :

A	Author's name
B	Title of book containing item
C	City of publication
D	Date
E	Editor of book containing item
G	Government (NTIS) ordering number
I	Issuer (publisher)
J	Journal name
K	Keys (for searching)
L	Label
M	Memorandum label
N	Issue number
O	Other information
P	Page(s) of article
R	Technical report reference
T	Title
V	Volume number
X	or
Y	or
Z	Information not used by refer

Refer output

The output of *refer* is a stream of string definitions, one for each of the fields of each reference, as shown below.

| .]-

```
.ds [A authors' names ...
.ds [T title ...
.ds [J journal ...
...
.][ type-number
```

The *refer* program, in general, does not concern itself with the significance of the strings. The different fields are treated identically by *refer*, except that the **X**, **Y** and **Z** fields are ignored (see the **-i** option of *mkey*) in indexing and searching. All *refer* does is select the appropriate citation, based on the keys. The macro package must arrange the strings so as to produce an appropriately formatted citation. In this process, it uses the convention that the '**T**' field is the title, the '**J**' field the journal, and so forth. The *refer* program does arrange the citation to simplify the macro package's job, however. The special macro `.-` precedes the string definitions and the special macro `.]` follows. These are changed from the input `.[` and `.]` so that running the same file through *refer* again is harmless.

The `.-` macro can be used by the macro package to initialize. The `.]` macro, which should be used to print the reference, is given an argument `type-number` to indicate the kind of reference, as follows :

- 1 Journal article
- 2 Book
- 3 Article within book
- 4 Technical report
- 5 Bell Labs technical memorandum
- 0 Other

The type is determined by the presence or absence of particular fields in the citation (a journal article must have a '**J**' field, a book must have an '**T**' field, and so forth). To a small extent, this violates the above rule that *refer* does not concern itself with the contents of the citation; however, the classification of the citation in *troff* macros would require a relatively expensive and obscure program. Any macro writer may, of course, preserve consistency by ignoring the argument to the `.]` macro. The reference is flagged in the text with the sequence

```
| \*[.number\*(.]
```

where `number` is the footnote number. The strings `.[` and `.]` should be used by the macro package to format the reference flag in the text. These strings can be replaced for a particular footnote. The footnote number (or other signal) is available to the reference macro `.]` as the string register `[F`. To simplify dealing with a text reference that occurs at the end of a sentence, *refer* treats a reference which follows a period in a special way. The period is removed, and the reference is preceded by a call for the string `<`, and followed by a call for the string `>`. For example, if a reference follows '*end.*' it will appear as

```
| end\*(<\*[.number\*(.)\*(>
```

where `number` is the footnote number. The macro package should turn either the string `>`, or `<`, into a period and delete the other one. This permits the output to have either the form '*end[31]*.' or '*end.31*' as the macro package wishes. Note that in one case the period precedes the number and in the other it follows the number.

In some cases users wish to suspend the searching, and merely use the reference macro formatting. That is, the user doesn't want to provide a search key between `.[` and `.]` brackets, but merely the reference lines for the appropriate document. Alternatively, the user can wish to add a few fields to those in the reference as in the standard file, or override some fields. Altering or replacing fields, or supplying whole references, is easily done by inserting lines beginning with `%`; any such line is taken as direct input to the reference processor rather than keys to be searched. Thus

```
.[
key1 key2 key3 ...
%Q New format item
%R Override report name
.]
```

makes the indicates changes to the result of searching for the keys. All of the search keys must be given before the first % line. If no search keys are provided, an entire citation can be provided in-line in the text. For example, if the eqn paper citation were to be inserted in this way, rather than by searching for it in the data base, the input would read

```
.[
%A B. W. Kernighan
%A L. L. Cherry
%T A System for Typesetting Mathematics
%J Comm. ACM
%V 18
%N 3
%P 151-157
%D March 1975
.]
```

This would produce a citation of the same appearance as that resulting from the file search. As shown, fields are normally turned into troff strings. Sometimes users would rather have them defined as macros, so that other troff commands can be placed into the data. When this is necessary, simply double the control character % in the data. Thus the input

```
.[
%V 23
%%M
Bell Laboratories,
Murray Hill, N.J. 07974
.]
```

is processed by refer into

```
.ds [V 23
.de [M
Bell Laboratories,
Murray Hill, N.J. 07974
..
```

The information after %%M is defined as a macro to be invoked by .[M while the information after %V is turned into a string to be invoked by *(V. At present -ms expects all information as strings.

License

The text of this manual page comes from *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk, which is distributed under the original BSD license.

See also

refer(1), sortbib(1), mkey(1), inv(1), hunt(1), and *Some application of Inverted Indexes in the UNIX System* by M. E. LESK.

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Authors

M. E. LESK. Modified by Pierre-Jean FICHET.

Sortbib manual

Name

sortbib – Sort refer bibliographic database following iso-690 standart

Synopsis

sortbib [-i|-sKEYS] database ...

Description

Sortbib sorts files of records containing *refer* key-letters by user-specified keys. Records may be separated by blank lines, or by .[and .] delimiters, but the two styles may not be mixed together. This program reads through each *database* and pulls out key fields, which are sorted separately. The sorted key fields contain the file pointer, byte offset, and length of corresponding records. These records are delivered using disk seeks and reads, so *sortbib* may not be used in a pipeline to read standard input.

By default, *sortbib* alphabetize the first %A and %D fields, which contain the senior author and date.

The -i option is used to sort database following an algorithm which aims to respect the *iso-690* bibliography standart :

- If J is defined, the order of fields is : QA+TE+SVBJ.
- Else, if B is defined, the order of fields is : QA+TE+SVB.
- Else, the order of fields is : QA+E+SVT.

The -s option is used to specify new KEYS. For instance, -sATD will sort by author, title, and date, while -sA+D will sort by all authors, and date. Sort keys past the fourth are not meaningful. No more than 16 databases may be sorted together at one time. Records longer than 4096 characters will be truncated.

Sortbib sorts on last word on the %A and %E line, which are assumed to be the author's and editor's last name. A word in the final position, such as "jr." or "ed.", will be ignored if the name beforehand ends with a comma. Authors with two-word last names or unusual constructions can be sorted correctly by using the *nroff* convention "\0" in place of a blank. A %Q field is considered to be the same as %A, except sorting begins with the first, not the last, word. *Sortbib* sorts on the last word of the %D line, usually the year. If a sort-significant field is absent from a record, *sortbib* places that record before other records containing that field.

License

Sortbib is distributed under the CDDL1.0 license. This manual page is distributed under a four clause BSD License.

Files

@BINDIR@/sortbib

See also

refer(1), u-ref(7).

Help and bugs

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

Authors

Written by Greg SHENAUT & Bill TUTHILL. Modified by Pierre-Jean FICHET.

Inv manual

Name

inv — Make an inverted index from output of mkey.

Synopsis

inv [-danpv] [-hn] [-i [u] name] outfile]

Description

The *inv* program computes the hash codes and writes the inverted files. It reads the output of *mkey* and writes the set of files described earlier in this section. It expects one argument, which is used as the base name for the three (or four) files to be written. Assuming an argument of *Index* (the default) the entry file is named *Index.ia*, the posting file *Index.ib*, the tag file *Index.ic*, and the key file (if present) *index.id*.

The *inv* program recognize the following options :

- a Append the new keys to a previous set of inverted files, making new files if there is no old set using the same base name.
- d Write the optional key file. This is needed when you can not check for false drops by looking for the keys in the original inputs, i.e. when the key derivation procedure is complicated and the output keys are not words from the input files.
- hn The hash table size is *n* (default 997); *n* should be prime. Making *n* bigger saves search time and spends disk space.
- i [u] name
Take input from file *name*, instead of the standard input; if **u** is present name is unlinked when the sort is started. Using this option permits the sort scratch space to overlap the disk space used for input keys.
- n Make a completely new set of inverted files, ignoring previous files.
- p Pipe into the sort program, rather than writing a temporary input file. This saves disk space and spends processor time.
- v Verbose mode; print a summary of the number of keys which finished indexing.

About half the time used in *inv* is in the contained sort. Assuming the sort is roughly linear, however, a guess at the total timing for *inv* is 250 keys per second. The space used is usually of more importance : the entry file uses four bytes per possible hash (note the **-h** option), and the tag file around 15-20 bytes per item indexed. Roughly, the posting file contains one item for each key instance and one item for each possible hash code; the items are two bytes long if the tag file is less than 65536 bytes long, and the items are four bytes wide if the tag file is greater than 65536 bytes long. To minimize storage, the hash tables should be over-full; for most of the files indexed in this way, there is no other real choice, since the entry file must fit in memory.

Files

@BINDIR@/invExecutable.

Assuming an argument of *Index* (the default) :

Index.ia Entry file.

Index.ib Posting file.

Index.ic Tag file.

Index.id Key file.

License

The text of this manual page comes from *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk, which is distributed under the original BSD license. The *inv* software is distributed under the CDDL-1.0 license.

See also

refer(1), referformat(7), mkey(1), hunt(1), and *Some application of Inverted Indexes in the UNIX System* by M. E. LESK.

Help and bugs

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

M. E. LESK. Modified by Pierre-Jean FICHET

Mkey manual

Name

Mkey – Build an index of keys from files

Synopsis

mkey [-sw] [-c *name*] [-f *name*] [-i *chars*] [-kn] [-ln] [-nm] *file* [*file...*]

Description

Mkey reads its input from the file names given as arguments, and, if there is no arguments, it reads from the standard input. It assumes that blank lines in the input delimit separate items for each of which a different line of **keys** should be generated. The lines of keys are written on the standard output. **Keys** are any alphanumeric string in the input not among the most frequent words in English and not entirely numeric (except that all-numeric strings are acceptable if they are between 1900 and 1999). In the output, keys are translated to lower case, and truncated to six characters in length; any associated punctuation is removed.

The following flags arguments are recognized by **mkey**.

- c *name* Name of file of common words. Default is `@LIBDIR@/eign`.
- f *name* Read a list of files from *name* and take each as an input argument.
- i *chars* Ignore each line which begins with '%' followed by any character in *chars*.
- kn Use at most *n* keys per input item. Unless this option is used, the output of **mkey** is comparable in size to its input.
- ln Ignore items shorter than *n* letter longs. Default is 3.
- nm Ignore as a key any word in the first *m* words of the list of common English words. The default is 100.
- s Remove the labels (*file :start,length*) from the output; just give the keys. Used when searching rather than indexing.
- w Each whole file is a separate item; blank lines are irrelevant.

Files

`@BINDIR@/mkeyExecutable`.

`@LIBDIR@/eign` List of frequent English words.

License

The text of this manual page comes from *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk, which is distributed under the original BSD license. The software *mkey* is distributed under the CDDL-1.0 license.

See also

refer(1), referformat(7), sortbib(1), inv(1), hunt(1), and *Some application of Inverted Indexes in the UNIX System* by M. E. LESK.

Help and bugs

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

Authors

This manual page is made from the document *Some application of Inverted Indexes in the UNIX System* by M. E. LESK. Modified by Pierre-Jean FICHET

Hunt manual

Name

hunt – Retrieve items from an *mkey* index.

SYNOPSIS

hunt [-agp] [-Cn] [-F[ynd]] [-i *string*] [-l n] [-o *string*] [-T[ynd]] [-t *string*] < *file*

Description

The *hunt* program retrieves items from an index. It combines, as mentioned above, the two parts of that process : search and delivery. The reason why it is efficient to combine delivery and search is partly to avoid starting unnecessary processes, and partly because the delivery operation must be a part of the search operation in any case. Because of the hashing, the search part takes place in two stages : first items are retrieved which have the right hash codes associated with them, and then the actual items are inspected to determine false drops, i.e. to determine if anything with the right hash codes doesn't really have the right keys. Since the original item is retrieved to check on false drops, it is efficient to present it immediately, rather than only giving the tag as output and later retrieving the item again. If there were a separate key file, this argument would not apply, but separate key files are not common.

Input to *hunt* is taken from the standard input, one query per line. Each query should be in *mkey* -s output format ; all lower case, no punctuation. The *hunt* program takes one argument which specifies the base name of the index files to be searched. Only one set of index files can be searched at a time, although many text files may be indexed as a group, of course. If one of the text files has been changed since the index, that file is searched with *fgrep* ; this may occasionally slow down the searching, and care should be taken to avoid having many out of date files.

The following option arguments are recognized by *hunt* :

- a Give all output ; ignore checking for false drops.
- Cn Coordination level *n* ; retrieve items with not more than *n* terms of the input missing ; default -C0 , implying that each search term must be in the output items.
- F[ynd] '-Fy' gives the text of all the items found ; '-Fn' suppresses them. '-Fd' where *d* is an integer gives the text of the first *d* items. The default is -Fy.
- g Do not use *fgrep* to search files changed since the index was made ; print an error comment instead.
- i *string* Take *string* as input, instead of reading the standard input.
- l n The maximum length of internal lists of candidate items is *n* ; default is 1000.
- o *string* Put text output ('Fy') in *string* ; of use only when invoked from another program.
- p Print hash code frequencies ; mostly for use in optimizing hash table sizes.
- T[ynd] '-Ty' gives the tags of the items found ; '-Tn' suppresses them. '-Td' where *d* is an integer gives the first *d* tags. The default is -Tn.
- t *string* Put tag output ('Ty') in *string* ; of use only when invoked from another program.

The timing of *hunt* is complex. Normally the hash table is overfull, so that there will be many false drops on any single term ; but a multi-term query will have few false drops on all terms. Thus if a query is underspecified (one search term) many potential items will be examined and

discarded as false drops, wasting time. If the query is overspecified (a dozen search terms) many keys will be examined only to verify that the single item under consideration has that key posted.

As would be expected, the optimal search is achieved when the query just specifies the answer; however, overspecification is quite cheap. In general, overspecification can be recommended; it protects the user against additions to the data base which turn previously uniquely-answered queries into ambiguous queries.

Files

@BINDIR@/huntExecutable.

License

The text of this manual page comes from *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk, which is distributed under the original BSD license. The software *hunt* is distributed under the CDDL-1.0 license.

See also

refer(1), referformat(7), sortbib(1), hunt(1), mkey(1), inv(1), and *Some application of Inverted Indexes in the UNIX System* by M. E. Lesk.

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Authors

M. E. LESK, modified by Pierre-Jean FICHET.

Tchars manual

Name

Tchars – Translate characters for Troff.

Synopsis

@BINDIR@/tchars [-e] *file*

Description

Tchars is a troff pre-processor which translate some ascii code to complex utf-8 characters. Actually, it can only translate a subset of betacode to polytonic greek.

Options

-e Escape characters : print troff utf-8 escape sequence in the form of `\U'xxxx'` instead of printing the utf-8 character itself.

Macros

In the troff document, characters that has to be translated must be indicated by one of the following macros :

CW *code after before*

 Translate *code*, and append *before* and *after* just before and after the translation.

CL *code* [*code...*]

 Translate the whole line following **CL**.

CS Begin a block that has to be translated. Must be followed by **CE**.

CE Close a block that has to be translated.

For *tchars* to work, you must define these macro to call a font which contain polytonic greek characters, as follow, for example :

```
.fp 0 RGN GFSPyrsos otf \ " Pyrsos from Greek Font Society
.de CW
\\$3\f[RGN]\\$1\fr\\$2
..
.de CL
\f[RGN]\\$*\fr
..
.de CS
.       ft RGN
..
.de CE
.       ft R
..
```

Code

A rune is composed of an ascii letter, followed by a representation of an accent. *Tchars* actually understand the following accents :

+ dialytika ¨
 & macron ¯
 ´ vrachy ˘
) psili ˙
 (dasia ˆ
 \ varia `
 / oxia ˆ
 = perispomeni ~
 | ypogegrammeni ,

The order is significant, it must be : +&')(V=|

Tchars actually understand the following letters :

A A ALPHA, a α alpha
 B B BETA, b β beta
 C Ξ XI, c ξ xi
 D Δ DELTA, d δ delta
 E Ε EPSILON, e ε epsilon
 F Φ PHI, f φ phi
 G Γ GAMMA, g γ gamma
 H Η ETA, h η eta
 I Ι IOTA, i ι iota
 K Κ KAPPA, k κ kappa
 Λ Λ LAMBDA, λ λ lambda
 Μ Μ MU, μ μ mu
 Ν Ν NU, ν ν nu
 Ο Ο OMICRON, ο ο omicron
 Π Π PI, π π pi
 Ρ Ρ RHO, ρ ρ rhô
 Σ Σ SIGMA, σ σ sigma
 Τ Τ TAU, τ τ tau
 Υ Υ UPSILON, υ υ upsilon
 Ω Ω OMEGA, ω ω omega
 Χ Χ CHI, χ χ chi
 Ψ Ψ PSI, ψ ψ psi
 Ζ Ζ ZETA, ζ ζ zeta

Example

| The word physis in polytonic greek is written:
 | .CW fu/sis .

Files

@BINDIR@/tchars

See also

utmac(7)¹.

License

¹ See p. 11.

Tchars and this man page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

Author

Written by Pierre-Jean FICHET.

Prexml and postxml manual

Name

prexml postxml – *nroff* processors to build xml files.

Synopsis

```
@BINDIR@/prexml < file.tr | nroff -mux | @BINDIR@/postxml > file.xml
```

Description

Postxml provide a solution to build an xml file from a troff document that rely on *nroff* itself. Since *nroff* build a plain text file from a troff document, it can be used, with an appropriate macro, to produce an intermediate file that *postxml* will translate to xml.

That intermediate file contain tags that *postxml* will transform to xml tags. If these tags are present in the source file, and need to be printed as is, *prexml* simply escape these tags so that *postxml* won't substitute them.

Since the final xml file might contain utf-8 characters, the **-Tlocale** option of *nroff* must be used.

prexml and *postxml* are provided with the **ux** macro that insert tags in an **utmac** document.

So, the command line could be :

```
| prexml < file.tr | nroff -Tlocale \  
| -mux | postxml > file.xml
```

Tags

Postxml

Postxml will substitute the following tags.

```
#[      Replaced by <.
#]      Replaced by >.
#(      Replaced by &.
#)      Replaced by ;.
#~      Replaced by #.
\n#-\n  Removed from stream.
```

So the following text :

```
| Some text
| #-
| #[a#]
| #-
| note text
| #[/a#]
| #(lt#).
```

Will be transform in :

! Some text<a>note text <.

Prexml

All these tags will be escaped by *prexml* if they are present in the original file, following these rules :

#[Replaced by #~[.
#] Replaced by #~].
#(Replaced by #~(.
#) Replaced by #~).
#~ Replaced by #~~.
\n#\n Replaced by #~-.

Files

@BINDIR@/prexml @BINDIR@/postxml

See also

nroff(1) , trxtr(1).

License

Prexml, *postxml* and this man page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Author

Written by Pierre-Jean FICHET.

Trxtr manual

Name

trxtr – Troff xml translator.

Synopsis

trxtr *stylesheet* < *infile.xml* > *outfile.xml*

Description

Trxtr is a simple shell script which use *xsltproc* to translate an xml file produced by the *ux* nroff macro and *postxml* to some other xml file. *trxtr* has been made to be used at the end of a nroff pipe :

```
| prexml < f.tr | nroff -Tlocale -mux | \  
| postxml | trxtr fodt > f.fodt
```

Current stylesheets are : *html*, for xhtml document; *fodt*, for flat open document text; and *utroff.org*, which is used to build the website utroff.org¹.

Files

@*BINDIR*/@*trxtr* the shell script; @*XSLDIR*/@*Directory* containing stylesheets.

See also

nroff(1), postxml(1), utmac.

License

Trxtr and this man page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at [help at utroff dot org](mailto:help@utroff.org). Please, send bugs and patches at [help at utroff dot org](mailto:help@utroff.org).

Author

Written by Pierre-Jean FICHET.

¹ <<http://utroff.org>>

Dpage manual

Name

Dpage – dpage puts two postscript page on a single sheet of paper.

Synopsis

@BINDIR@dpage file

Description

Dpage puts two postscript page produced by *Heirloom dpost* on a single sheet of paper. Actually, it can only put two A4 page on an A3 page. *Dpage* don't transform the size of the pages. It can be used after *psbook* to produce booklets.

Example

To produce a booklet from a troff file, you can use the following command line :

```
| troff f.tr | dpost | psbook -s16 | dpage | ps2pdf - f.pdf
```

Once done, you just have to print the document, fold the sheets, and bind your book.

Files

@BINDIR@dpage

See also

dpost(1), psbook(1)

License

Dpage and this man page are distributed under a two clause BSD license.

Help and bugs

Don't hesitate to ask questions at help at utroff dot org. Please, send bugs and patches at help at utroff dot org.

Author

Written by Pierre-Jean FICHET.

Changes of Utroff

2014/03/17 13 :37 :01 utroff-0.3

- config.mk : Update release to 0.3. Fix SSHPUB variable.
- www : doc.tr : Add U documentation. Fix documentation links.
- U : Remove bashisms (bug report by fravashyo) and fix a bug in the directory list of fonts.

2014/03/13 09 :54 :26 utroff-0.2

- all : Add a makefile to build heirloom doctools easily. Add the U script to use utroff easily.
- demo : Set a minimal troff environment to build demo files.
- frog : dlog.sh now work with subdirectories.
- refer : Fix a link to license in documentation.
- share : Add 2013 and 2014 year in bsd2 license. Fix links in utroff manual page.
- tchars : RFC 3629 ends utf8 at U+10FFFF, so that utf8 is never encoded on more than 4 bits (bug report by seb).
- ugrind : Fix link to license in documentation.
- utmac : Load a font if the user locale macro does not do it (bug report by Fravashyo). Avoid arithmetic overflow in macro uh. Use hardcoded path for the refer command to be sure to use utroff refer.
- www : The html log page now only print a summary of changes. Add the OFL license of font-awesome. Add log page for demo, U and share.
- build system : Improve target pub and clean. Define a minimal troff environment when using troff and nroff. Substitute variables while building shell scripts. Add RELEASE variable, to get a simpler build system. The utroff readme is now on the share directory. Look for only one config file (config.mk). Create an md5sum for each archive.

2013/11/05 17 :51 :54 utroff-0.1

- Initial revision.
-

Changes of U

2014/03/17 13 :23 :48 VERSION-0.4

- U.sh : Fix Remove bashisms (bug report by fravashyo).
- U.sh : Fix colon separated list of directory.
- U.sh : Fix licence date.

2014/03/13 09 :46 :29 VERSION-0.3

- U.tr : Fix typo, since u has been renamed U.

2014/03/12 13 :59 :49 VERSION-0.2

- all : rename u to U, to avoid conflicts with plan9 u.

2013/11/14 12 :16 :52 VERSION-0.1

- Initial revision.
-

Changes of utmac

2014/03/13 09 :47 :07 VERSION-0.13

- u-ref.tmac : Fix refer command to be sure to use Utroff refer.

2014/03/12 20 :08 :14 VERSION-0.12

- uh.tmac : Add unit to avoid arithmetic overflow.

2014/03/12 14 :04 :10 VERSION-0.11

- uh.tmac, us.tmac : FIX : load font if user locale macro does not do it (bug report by Fravashyo).

2013/11/04 14 :42 :31 VERSION-0.10

- uh.tmac : FIX : NT and NB notes now have the same size as normal notes, since it's much more coherent.

2013/11/04 09 :03 :16 VERSION-0.9

- uh.tmac, us.tmac, ut.tmac, ux.tmac : FIX : before printing its arguments, the var :print macro now print a '&' to avoid problems with initials '.
- uw.tmac : FIX : code begins by """" and not by ""'".

2013/10/30 20 :34 :24 VERSION-0.8

- troff files : fix link to license.
- makefile : license is bsd2.

2013/10/30 15 :20 :07 VERSION-0.7

- makefile : FIX : Use new build system.
- all : FIX : recognize correctly if we are inside a note or not.
- u-apolline.tmac : FIX : escape font definition by using "\E" to may use fonts as macro argument.
- u-fr.tmac : FIX : French abbreviation of "collection" is "coll.", and "Index des mots" is now "Index analytique", to follow the french usage.
- uh.tmac : FIX : NT and NB notes use a bigger font size, since they are used to give important informations.
- uh.tmac : FIX : Recto-verso now have a better support.
- uh.tmac : FIX : H1 heading is larger (32 instead of 24).
- uh.tmac : FIX : use \en bullet in lists instead of \em.
- uh.tmac : FIX : .PI macro does a more subtle job to print the bullet : default bullet is printed with an indent of 2v, user bullet is printed with an indent of 0v, if user bullet is longer than 3v the text of the paragraph begins on the next line, otherwise, it begins on the same line, with an indent of 3v (bug report by Frédéric Mazaoit).
- u-idx.tmac : FIX : Indexes use their own paragraph method.
- u-idx.tmac : FIX : Print correctly the list of names with .LN macro.
- u-idx.tmac : FIX : do not use absolute file name while to call the idx binary.
- u-idx.tmac : FIX : inside a note, do not print url if some text is defined by LP. Idem for mail adress.
- u-libertine.tmac : FIX : escape font definition by using "\E" to may use fonts as macro argument.
- u-ref.tmac : FIX : in bibliography lists when books are from the same author the line is

now printed correctly.

- u-ref.tmac : FIX : Print H0 heading.
- u-ref.tmac : FIX : do not use absolute file name to launch the refer binary.
- uw.tmac : FIX : Escape font definition with \E to be sure that they are not interpreted in copy mode, and so can be used in arguments of macros.

2013/04/10 16 :21 :10 VERSION-0.6

- All : Add US macro to test for used macro.
- uh.tmac, us.tmac : Insert a keyword on the next heading if filename change. Be sure that the keyword is inserted only once.
- u-idx.tmac : Inform user if a keyword as yet been inserted.
- u-grind.tmac : Use a darker line for code.
- u-grind.tmac : Do not add space after block of code.
- um.tmac : Don't print " :" in H2 macro.
- u-ref.tmac : The size of the dash in bibliography list is shorter to give better result with the new us.tmac .PI macro.
- us.tmac : Ornament is no longer printed (bug report by Frédéric Mazoit).
- us.tmac : Give correct size to note trap.
- us.tmac : Insert a space of 1.5 before H3 macro and 0.5 after (bug report by Frédéric Mazoit).
- us.tmac : Implement recto-verso header, as documented in manual page (bug report by Frédéric Mazoit).
- us.tmac : Be sure that only two lines are skipped after H2 macro.
- us.tmac : PI macro now does a more subtle job to print the bullet : default bullet is printed with an indent of 2v, and text is printed with an indent of 3v, on the same line ; an user defined bullet is printed without any indent, and text is printed on the same line with an indent of 3v if the bullet is shorter than 3v, or in the next line if the bullet is longer than 3v (bug report by Frédéric Mazoit).

2013/04/07 17 :46 :46 VERSION-0.5

- makefile : s/ROOT/DESTDIR/g (bug report by Seb).
- all : Add PB macro, for left extended paragraph. PL macro is now for left align paragraph.
- um, ut, uw, ux : Link unused macros to a null macro to avoid error messages.
- um, ut, uw : Add all index macros.
- um, ut, uw, ux : add data-tmac string.
- ut : Notes are now diverted to the bottom of the output file.
- uw.tmac : Load the ut macro instead of rewriting everything, and just define differences.
- ux.tmac : Close tag before .vS macro (bug report by Seb).

2013/04/04 16 :48 :43 VERSION-0.4

- All : Utmac now use a better syntax for user macros : Appendix macros now begin by an X ; and inline macros now begin by an L.
- ux.tmac : Lists are now enclosed in <PLIST> tag, since LI is now used by inline macros.
- ut.tmac : lots of bug fix.
- ut.tmac : Use u-idx to define inline macros. This is still a work in progress, since the page number is not relevant.
- u-en : Typo bug fix : print "widow" and not "wido".

2013/03/31 08 :35 :53 VERSION-0.3

- All : Add DI macro to define id.
- All : All locale strings are now defined. Macros do not use hardcoded strings anymore.
- um.tmac, ux.tmac, uw.tmac : Use warnings by default.
- us.tmac (0.4) : Change the layout : print an header, header and footer draw a bold line, notes NB and NT are part of the flow of the text.
- Fonts : remove some non existing ligatures.

- u-libertine.tmac : Define IC, IA, IU, IF fonts (bug fix).
- u-apolline.tmac : Do not use monospace font anymore for code, since none is mounted.
- u-grind.tmac : Use less colors, and draw the left border line by using a more robust system than lpdfx.
- u-idx.tmac : Use special support for names, in the manner of refer (firstname lastname, jr).
- u-idx.tmac : use idx -d option to delete \004.
- u-idx.tmac : test if inside a note before printing a note.
- ux.tmac : Add PC macro.
- ux.tmac : use <UTMAC> instead of <HLM> for root.
- ut.tmac : Fix various non significant bugs, but which made warnings.
- u-ref.tmac : use refer %K key as keyword instead of %L key. The %K is needed to look for references yet cited (op. cit. system).

2013/03/24 21 :04 :22 VERSION-0.2

- ut.tmac : .PI does not print "txt :print" anymore.

2013/03/24 17 :32 :10 VERSION-0.1

- Initial revision.
-

Changes of idx

2013/10/30 20 :14 :38 VERSION-0.6

- idx.tr, readme.tr : fix link to license.

2013/10/30 16 :25 :14 VERSION-0.5

- FIX : idx renamed idx.sh for a simpler build method.

2013/10/30 14 :54 :16 VERSION-0.4

- makefile : use new build system.

2013/04/07 17 :35 :08 VERSION-0.3

- idx : Use case ... esac for arguments (advise by Seb).
- idx : Avoid absolute name for sort and awk (bug report by Seb).
- makefile : FIX : s/ROOT/DESTDIR/g (bug report by Seb).

2013/04/04 16 :31 :18 VERSION-0.2

- idx : Do not print anything if idx file is empty when converting to troff.

2013/03/30 21 :43 :59 VERSION-0.1

- Initial revision.
-

Changes of soelim

2013/10/30 20 :22 :14 VERSION-0.6

- troff files : fix link to license.

2013/10/30 16 :30 :27 VERSION-0.5

- makefile : FIX : scripts are in BIN variables. Files to strip are in STR variable.

2013/10/30 15 :02 :56 VERSION-0.4

- makefile : Use new build system.

2013/04/10 16 :07 :53 VERSION-0.3

- Soelim : Use allocated memory instead of destroying and recreating it for each ".so" line (bug report by Seb).

2013/04/07 17 :38 :57 VERSION-0.2

- makefile : s/ROOT/DESTDIR/g (bug report by Seb).
- soelim.c : Use malloc instead of array to define file name to fix segfault if filename is longer than 100 char (bug report by Seb).

2013/03/24 17 :15 :44 VERSION-0.1

- Initial revision.
-

Changes of ugrind

2014/03/12 14 :29 :56 VERSION-0.15

- version.c : troff files : fix link to license.

2013/10/30 20 :29 :48 VERSION-0.14

- troff files : fix link to license.

2013/10/30 16 :35 :09 VERSION-0.13

- makefile : FIX : scripts are in BIN variables. Files to strip are in STR variable.

2013/10/30 16 :33 :47 VERSION-0.12

- makefile : FIX : scripts are in BIN variables. Files to strip are in STR variable.

2013/10/30 15 :07 :56 VERSION-0.11

- makefile : Use new build system.

2013/04/10 16 :01 :19 VERSION-0.10

- Troff language : print "" instead of "." if macro begins by "".

2013/04/07 17 :40 :27 VERSION-0.9

- makefile : s/ROOT/DESTDIR/g (bug report by Seb).

2013/04/05 10 :22 :33 VERSION-0.8

- Troff language : Bugfix, close properly "\n" and "\".

2013/03/24 17 :29 :19 VERSION-0.7

- Use new name : Ugrind.

2013/03/22 11 :09 :37 VERSION-0.6

- ADD : makefile.

2013/03/22 11 :07 :57 VERSION-0.5

- ADD : hardcoded support for troff language.
- FIX : -l option force whole file mode.

2013/03/22 11 :04 :26 VERSION-0.4

- FIX : filter mode is now default.
- DEL : two column mode, font size, and index options are deleted, and hardcoded typographic tricks, since that the job of the macro to handle this.
- FIX : tabstop is 4 by default.
- ADD : The format of the grindex file is now documented, in comments of the file.
- ADD : make, postxml, xml, refer and idx languages are added to grindex file.

2013/03/22 11 :02 :20 VERSION-0.3

- FIX : Files are renamed : Vfontedpr is now grind, vgrindex is now grindex, vgrindex.c is now grindex.c.
 - FIX : Grind now makes a difference between begin and end of a match, and so, colorize code more precisely.
-

- FIX : in gridefs, \p now behaves less strangely.

2013/03/22 11 :00 :53 VERSION-0.2

- ADD : Portions copyright and new RCS Id.

2013/03/22 10 :58 :52 VERSION-0.1

- Initial revision.
-

Changes of refer

2014/03/12 14 :25 :26 VERSION-0.16

- version.c : troff files : fix link to license.

2013/10/30 20 :18 :17 VERSION-0.15

- troff files : fix link to license.

2013/10/30 16 :27 :50 VERSION-0.14

- makefile : FIX : scripts are in BIN variables. Files to strip are in STR variable.

2013/10/30 14 :58 :23 VERSION-0.13

- makefile : FIX : Use new build system.

2013/04/07 17 :37 :35 VERSION-0.12

- makefile : s/ROOT/DESTDIR/g (bug report by Seb).

2013/03/31 07 :51 :24 VERSION-0.11

- makefile, refer : various typo bug fix.
- refer : close properly firstname small capital.

2013/03/22 11 :27 :55 VERSION-0.10

- Add makefile.

2013/03/22 11 :26 :33 VERSION-0.9

- refer : strings to define small capitals don't need to be escaped with "\".

2013/03/22 11 :25 :19 VERSION-0.8

- refer : do not add ";" after lastname if there's no firstname when reversing.

2013/03/22 11 :24 :16 VERSION-0.7

- refer : use string "&" to join names instead of "and" and "AND".

2013/03/22 11 :23 :06 VERSION-0.6

- refer : add support for editor names in the same way as author names (multiple editor names, small capitals...).

2013/03/22 11 :21 :46 VERSION-0.5

- refer : don't truncate utf8 chars when copying strings in comments of file.

2013/03/22 11 :19 :03 VERSION-0.4

- refer : add -d option to use strings to define small capitals instead of hardcoded fake small capitals. Strings are : .

2013/03/22 11 :17 :34 VERSION-0.3

- refer, sortbib : add -i option to sort following iso-690 standart.

2013/03/22 11 :16 :26 VERSION-0.2

- ADD : portions copyright and new RCS Id.
-

2013/03/22 11 :15 :15 VERSION-0.1

- Initial revision, from Heirloom Refer.



Changes of tchars

2013/11/07 17 :17 :56 VERSION-0.6

- tchars.c : FIX : RFC 3629 ends utf8 at U+10FFFF, so that utf8 is never encoded on more than 4 bits (bug report by seb).

2013/10/31 07 :47 :11 VERSION-0.5

- tchars.tr : add definition of the macros CW, CL, CS and CE as example.

2013/10/30 20 :26 :00 VERSION-0.4

- readme.tr, tchars.tr : fix link to license.

2013/10/30 16 :31 :46 VERSION-0.3

- makefile : FIX : scripts are in BIN variables. Files to strip are in STR variable.

2013/10/30 15 :05 :17 VERSION-0.2

- makefile : use new build system.
- all : common files are now in ../share directory
- tchars.c : ADD : comments for hexatochars.

2013/07/11 20 :20 :36 VERSION-0.1

- Initial revision.
-

Changes of troffxml

2013/10/30 20 :27 :47 VERSION-0.9

- troff files : fix link to license.

2013/10/30 16 :32 :24 VERSION-0.8

- makefile : FIX : scripts are in BIN variables. Files to strip are in STR variable.

2013/10/30 15 :06 :10 VERSION-0.7

- makefile : Use new build system.
- utroff.org.xsl : Use @PUBURL@ variable instead of @TUXFAM@.

2013/04/07 18 :01 :40 VERSION-0.6

- makefile : s/ROOT/DESTDIR/g (bug report by Seb).
- utofodt.xsl, utohtml.xsl, utroff.org.xsl : Add definition for PB tag (left extended paragraph). PL tag now means left align paragraph.
- utohtml.xsl, utroff.org.xsl : Define xhtml strict doctype (bug report by Seb).

2013/04/05 10 :26 :17 VERSION-0.5

- utroff.org.css, utroff.org.xsl : Use new awesome font, with F and T characters.
- utroff.org.css : small bug and typo fix.

2013/04/04 17 :10 :54 VERSION-0.4

- xsl files : use <UTMAC> as root instead of <HLM>.
- utroff.org.css : Add p.centered.
- utroff.org.css, utroff.org.xsl : Use @TUXFAM@ to define url for tuxfamily download repository.
- utroff.org.xsl : Always define a title, even if DT is not defined.
- utroff.org.xsl : Transform <DI> to <meta Id="...">
- utroff.org.xsl : Various typo fix.
- makefile : chmod 644 xsl and css files.
- utroff.org.css : Add font support for various browsers, including firefox which does not allow cross domain in css.

2013/03/31 19 :26 :57 VERSION-0.3

- xsl files : use <UTMAC> as root instead of <HLM>.
- utroff.org.css : Add p.centered.
- utroff.org.css, utroff.org.xsl : Use @TUXFAM@ to define url for tuxfamily download repository.
- utroff.org.xsl : Always define a title, even if DT is not defined.
- utroff.org.xsl : Transform <DI> to <meta Id="...">
- utroff.org.xsl : Various typo fix.
- makefile : chmod 644 xsl and css files.
- utroff.org.css : Add local font for firefox which does not allow cross domain in css.

2013/03/24 17 :26 :49 VERSION-0.2

- utroff.org.xsl, utroff.org.css : use new website url.
 - utroff.org.xsl : add flattr and twitter links.
-

2013/03/22 16 :43 :47 VERSION-0.1

- Initial revision.
-

Changes of dpage

2013/10/30 20 :11 :05 VERSION-0.4

- FIX : link to license.

2013/10/30 16 :23 :25 VERSION-0.3

- makefile : FIX : scripts are in BIN variables. Files to strip are in STR variable.

2013/10/30 14 :49 :18 VERSION-0.2

- all : common files are now in ../share directory.
- makefile : Include ../config.mk

2013/07/14 10 :27 :28 VERSION-0.1

- Initial revision.
-

Two clause BSD License

Copyright (c) 2012-2014

Pierre-Jean Fichet. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met :

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Original BSD License

Copyright (c) 1980, 1993

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met :

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement : This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Common Development and Distribution License Version 1.0

1. Definitions

1.1.

"Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2.

"Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3.

"Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4.

"Executable" means the Covered Software in any form other than Source Code.

1.5.

"Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6.

"Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7.

"License" means this document.

1.8.

"Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9.

"Modifications" means the Source Code and Executable form of any of the following :

- A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications ;
- B. Any new file that contains any part of the Original Software or previous Modifications ;
or
- C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10.

"Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11.

"Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12.

"Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13.

"You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.**2.1. The Initial Developer Grant.**

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license :

- (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work ; and
- (b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
- (c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
- (d) Notwithstanding Section 2.1(b) above, no patent license is granted : (1) for code that You delete from the Original Software, or (2) for infringements caused by : (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license :

- (a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work ; and
- (b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of : (1) Modifications made by that Contributor (or portions thereof) ; and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
- (c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
- (d) Notwithstanding Section 2.2(b) above, no patent license is granted : (1) for any code that Contributor has deleted from the Contributor Version ; (2) for infringements caused by :

(i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You : (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. Disclaimer of Warranty.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. Termination.

6.1.

This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2.

If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3.

In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. *Limitation of Liability.*

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. *U.S. Government End Users.*

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. *Miscellaneous.*

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. *Responsibility for Claims.*

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

Notice Pursuant to Section 9 of the Common Development and Distribution License (CDDL)

For Covered Software in this distribution, this License shall be governed by the laws of the State of California (excluding conflict-of-law provisions).

Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

Tables

Table of content

Utroff	1
Utroff readme	2
Presentation of Utroff	2
Quick start	2
Dependencies	2
Configuration	2
Build and install	2
Environment variables	2
Conflicts	3
Makefile options	3
About the pre-built archive	4
About the development archive	4
Help and bugs	5
Utroff manual	6
Name	6
Description	6
Utroff session	6
Configure your environment	6
Think	6
Write	6
Pre-process	7
Format	7
Post-process	7
Recover your environment	7
Example	7
Environment	7
See also	8
License	8
Help and bugs	8
Author	8
U manual	9
Name	9
Synopsis	9
Description	9
Example	9
See also	10
License	10
Help and bugs	10
Author	10
Utmac manual	11
Name	11
Description	11
Files	11
User macro files	11
Locale macro files	11
Internal macro files	11

Macros	11
Metadata macros	12
Register macro to define format and layout	12
Utmac tests macro	12
Heading macros	13
Summaries macros	13
Paragraph macros	13
Typographic macros	13
Note macros	13
Inline macros	14
Appendix macros	14
Fonts	15
Naming conventions	15
Example	15
Environment	16
Files	16
See also	16
License	17
Help and bugs	17
Author	17
U-ref manual	18
Name	18
Description	18
Usage	18
Macros	18
Example	18
Fields	19
Files	20
See also	20
License	20
Help and bugs	20
Author	20
Idx manual	21
Name	21
Synopsis	21
Description	21
Options	21
Format	21
List of pages	21
List of words	21
Troff output	22
Troff input	22
Example	22
Compatibility	23
Files	23
See also	23
License	23
Help and bugs	23
Author	23
Soelim manual	24
Name	24
Synopsis	24
Description	24
See also	24
Notes	24
License	24

Help and bugs	24
Authors	25
Felim manual	26
Name	26
Synopsis	26
Description	26
See also	26
License	26
Help and bugs	26
Author	26
Slelim manual	27
Name	27
Synopsis	27
Description	27
See also	27
License	27
Help and bugs	27
Author	27
Ugrind manual	28
Name	28
Synopsis	28
Description	28
Programming style	28
Macros and strings	29
Macros	29
Strings	29
History	29
License	29
Files	29
See also	29
Help and bugs	29
Authors	30
Refer manual	31
Name	31
Synopsis	31
Description	31
Example	32
Files	32
See also	32
Notes	33
License	33
Help and bugs	33
Authors	33
Refer input and output format	34
Refer file	34
Refer output	34
License	36
See also	36
Help and bugs	36
Authors	37
Sortbib manual	38
Name	38
Synopsis	38
Description	38
License	38
Files	38

See also	39
Help and bugs	39
Authors	39
Inv manual	40
Name	40
Synopsis	40
Description	40
Files	40
License	41
See also	41
Help and bugs	41
Mkey manual	42
Name	42
Synopsis	42
Description	42
Files	42
License	42
See also	42
Help and bugs	43
Authors	43
Hunt manual	44
Name	44
SYNOPSIS	44
Description	44
Files	45
License	45
See also	45
Help and bugs	45
Authors	45
Tchars manual	46
Name	46
Synopsis	46
Description	46
Options	46
Macros	46
Code	46
Example	47
Files	47
See also	47
License	47
Help and bugs	48
Author	48
Prexml and postxml manual	49
Name	49
Synopsis	49
Description	49
Tags	49
Postxml	49
Prexml	50
Files	50
See also	50
License	50
Help and bugs	50
Author	50
Trxtr manual	51
Name	51

Synopsis	51
Description	51
Files	51
See also	51
License	51
Help and bugs	51
Author	51
Dpage manual	52
Name	52
Synopsis	52
Description	52
Example	52
Files	52
See also	52
License	52
Help and bugs	52
Author	52
Changes of Utroff	53
2014/03/17 13 :37 :01 utroff-0.3	53
2014/03/13 09 :54 :26 utroff-0.2	53
2013/11/05 17 :51 :54 utroff-0.1	53
Changes of U	54
2014/03/17 13 :23 :48 VERSION-0.4	54
2014/03/13 09 :46 :29 VERSION-0.3	54
2014/03/12 13 :59 :49 VERSION-0.2	54
2013/11/14 12 :16 :52 VERSION-0.1	54
Changes of utmac	55
2014/03/13 09 :47 :07 VERSION-0.13	55
2014/03/12 20 :08 :14 VERSION-0.12	55
2014/03/12 14 :04 :10 VERSION-0.11	55
2013/11/04 14 :42 :31 VERSION-0.10	55
2013/11/04 09 :03 :16 VERSION-0.9	55
2013/10/30 20 :34 :24 VERSION-0.8	55
2013/10/30 15 :20 :07 VERSION-0.7	55
2013/04/10 16 :21 :10 VERSION-0.6	56
2013/04/07 17 :46 :46 VERSION-0.5	56
2013/04/04 16 :48 :43 VERSION-0.4	56
2013/03/31 08 :35 :53 VERSION-0.3	56
2013/03/24 21 :04 :22 VERSION-0.2	57
2013/03/24 17 :32 :10 VERSION-0.1	57
Changes of idx	58
2013/10/30 20 :14 :38 VERSION-0.6	58
2013/10/30 16 :25 :14 VERSION-0.5	58
2013/10/30 14 :54 :16 VERSION-0.4	58
2013/04/07 17 :35 :08 VERSION-0.3	58
2013/04/04 16 :31 :18 VERSION-0.2	58
2013/03/30 21 :43 :59 VERSION-0.1	58
Changes of soelim	59
2013/10/30 20 :22 :14 VERSION-0.6	59
2013/10/30 16 :30 :27 VERSION-0.5	59
2013/10/30 15 :02 :56 VERSION-0.4	59
2013/04/10 16 :07 :53 VERSION-0.3	59
2013/04/07 17 :38 :57 VERSION-0.2	59
2013/03/24 17 :15 :44 VERSION-0.1	59
Changes of ugrind	60
2014/03/12 14 :29 :56 VERSION-0.15	60

2013/10/30 20 :29 :48 VERSION-0.14	60
2013/10/30 16 :35 :09 VERSION-0.13	60
2013/10/30 16 :33 :47 VERSION-0.12	60
2013/10/30 15 :07 :56 VERSION-0.11	60
2013/04/10 16 :01 :19 VERSION-0.10	60
2013/04/07 17 :40 :27 VERSION-0.9	60
2013/04/05 10 :22 :33 VERSION-0.8	60
2013/03/24 17 :29 :19 VERSION-0.7	60
2013/03/22 11 :09 :37 VERSION-0.6	60
2013/03/22 11 :07 :57 VERSION-0.5	60
2013/03/22 11 :04 :26 VERSION-0.4	60
2013/03/22 11 :02 :20 VERSION-0.3	60
2013/03/22 11 :00 :53 VERSION-0.2	61
2013/03/22 10 :58 :52 VERSION-0.1	61
Changes of refer	62
2014/03/12 14 :25 :26 VERSION-0.16	62
2013/10/30 20 :18 :17 VERSION-0.15	62
2013/10/30 16 :27 :50 VERSION-0.14	62
2013/10/30 14 :58 :23 VERSION-0.13	62
2013/04/07 17 :37 :35 VERSION-0.12	62
2013/03/31 07 :51 :24 VERSION-0.11	62
2013/03/22 11 :27 :55 VERSION-0.10	62
2013/03/22 11 :26 :33 VERSION-0.9	62
2013/03/22 11 :25 :19 VERSION-0.8	62
2013/03/22 11 :24 :16 VERSION-0.7	62
2013/03/22 11 :23 :06 VERSION-0.6	62
2013/03/22 11 :21 :46 VERSION-0.5	62
2013/03/22 11 :19 :03 VERSION-0.4	62
2013/03/22 11 :17 :34 VERSION-0.3	62
2013/03/22 11 :16 :26 VERSION-0.2	62
2013/03/22 11 :15 :15 VERSION-0.1	63
Changes of tchars	64
2013/11/07 17 :17 :56 VERSION-0.6	64
2013/10/31 07 :47 :11 VERSION-0.5	64
2013/10/30 20 :26 :00 VERSION-0.4	64
2013/10/30 16 :31 :46 VERSION-0.3	64
2013/10/30 15 :05 :17 VERSION-0.2	64
2013/07/11 20 :20 :36 VERSION-0.1	64
Changes of troffxml	65
2013/10/30 20 :27 :47 VERSION-0.9	65
2013/10/30 16 :32 :24 VERSION-0.8	65
2013/10/30 15 :06 :10 VERSION-0.7	65
2013/04/07 18 :01 :40 VERSION-0.6	65
2013/04/05 10 :26 :17 VERSION-0.5	65
2013/04/04 17 :10 :54 VERSION-0.4	65
2013/03/31 19 :26 :57 VERSION-0.3	65
2013/03/24 17 :26 :49 VERSION-0.2	65
2013/03/22 16 :43 :47 VERSION-0.1	66
Changes of dpage	67
2013/10/30 20 :11 :05 VERSION-0.4	67
2013/10/30 16 :23 :25 VERSION-0.3	67
2013/10/30 14 :49 :18 VERSION-0.2	67
2013/07/14 10 :27 :28 VERSION-0.1	67
Two clause BSD License	68
Original BSD License	69
Common Development and DiStribution License Version 1.0	70

1. Definitions	70
1.1.	70
1.2.	70
1.3.	70
1.4.	70
1.5.	70
1.6.	70
1.7.	70
1.8.	70
1.9.	70
1.10.	70
1.11.	71
1.12.	71
1.13.	71
2. License Grants.	71
2.1. The Initial Developer Grant.	71
2.2. Contributor Grant.	71
3. Distribution Obligations.	72
3.1. Availability of Source Code.	72
3.2. Modifications.	72
3.3. Required Notices.	72
3.4. Application of Additional Terms.	72
3.5. Distribution of Executable Versions.	72
3.6. Larger Works.	72
4. Versions of the License.	72
4.1. New Versions.	73
4.2. Effect of New Versions.	73
4.3. Modified Versions.	73
5. Disclaimer of Warranty.	73
6. Termination.	73
6.1.	73
6.2.	73
6.3.	74
7. Limitation of Liability.	74
8. U.S. Government End Users.	74
9. Miscellaneous.	74
10. Responsibility for Claims.	74
Notice Pursuant to Section 9 of the Common Development and Distribution License (CDDL)	75
Tables	76
Table of content	76
