Chapter 39: Creating and Formatting Man Pages

In this chapter we show you how to create man pages, format them, and even create html documents from them. This is not a comprehensive man page reference, but it contains sufficient information for most purposes.

For further information, from the **UNIX Resources** Web page, see “How to Create Man Pages” under *Software Development*.

☞ First, a few notes:

- The man pages for a **UPS** product can go anywhere, as long as the location is specified in the table file. A recommended location is `${UPS_PROD_DIR}/man` for the formatted pages and `${UPS_PROD_DIR}/catman` for the unformatted pages. The **UPS** backwards-compatible default, however, is `${UPS_UPS_DIR}/toman/man` for the formatted pages and `${UPS_UPS_DIR}/toman/catman` for the unformatted pages.
- Man page file names should consist of the product name, a period, and the section number as described in the following note. This applies to both formatted and unformatted files, which are distinguished by residing in separate directories.
- Man pages for commands are generally maintained as section 1, and library and system calls as section 3. The section number should appear as an extension of the man page file name (e.g., `hello.1` for the command **hello**). Here is a full listing of categories by section:

1	user commands
2	system calls
3	C library functions (on some platforms 3c for C, 3f for FORTRAN, etc.)
4	devices and network interfaces
5	file formats
6	games and demos
7	environments, tables, and **troff** macros
8	maintenance commands
9	x window system

l local commands

n new commands (**tcl** and **tk** use this)

- We recommend using either of the utilities **nroff** or **groff** with the **`-man`** option to format your man pages in a standard way. These utilities are documented in many standard UNIX texts, and you can also find man pages for them.

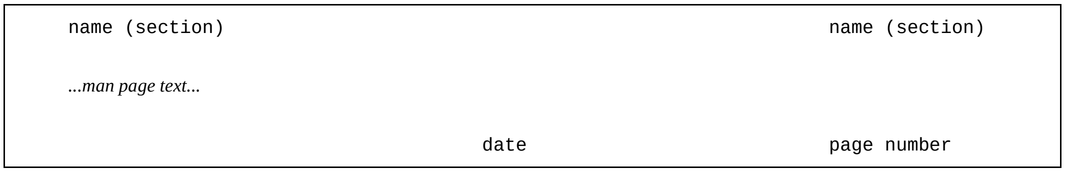
39.1 Creating the Source Document (Unformatted)

39.1.1 Source File Format

We recommend writing man pages in the source form using simple macros from the **nroff** macro package **`-man`**. Most of these macros require a dot (**`.`**) in the first column. The following list of macros is sufficient for writing standard man pages:

`.TH <name> <section> <date>`

Title Heading; specify product name, man page section (usually **`1`**), and date, in this order, to produce a man page format of this type:



```
name (section)                                           name (section)

...man page text...

                              date                       page number
```

.SH "<text>" Section Heading; if no blanks in text, quotes are not needed.

.SS "<text>" Subsection Heading; if no blanks in text, quotes are not needed.

.P Paragraph break

.IP "<item>" Starts an indented paragraph where “item” is put to the left of it; if no blanks in “item”, quotes are not needed.

.HP Starts a paragraph with a hanging indent; i.e. lines after the first are indented

.RE Defines an indented region

.B "<text>" Bold; if no blanks in text, quotes are not needed.

.I "<text>" Italic; this shows up as underlined on most terminals. If no blanks in text, quotes are not needed.

.TP <columns> Term/paragraph format; columns specify how many columns to allocate to the term column. As an example, this input:

```
.TP 5
f1
is one option
.TP
f2
is another option
```

produces this output under **nroff -man**:

```
     f1   is one option

     f2   is another option
```

where “`is`” starts in column 6. Notice that the first **.TP** sets the column value of the term, and the second one picks it up.

.P New paragraph

.br Break line

.nf Nofill (used to suppress normal line filling; used for preformatted text)

`.fi`	Fill (used to resume normal line filling, usually after a **`.nf`**)
`./"`	Comment line

39.1.2 Man Page Information Categories

Categories of information that you may want to include as section headings (**`.SH`**) are:

NAME

This should be the product name followed by a short description. The text on this line is also used as the keyword list for **`man -k`** and **`apropos`**.

SYNOPSIS or SYNTAX

Document here the complete syntax of the command used to invoke the product.

AVAILABILITY

Document here the OS flavors for which the program is available.

DESCRIPTION

Document here a full but succinct description of the use of the product.

OPTIONS

Document here all the options available for the invoking command.

EXAMPLES

Document here situations in which the program can be used, if there are uses that are not obvious.

NOTES

Document here any information the user should be aware of when using the command.

MESSAGES AND EXIT CALLS

Document here all errors and other messages returned to the user. Include the cause and the recovery actions whenever appropriate and possible.

AUTHOR

Document here the product coordinator and/or the major developers and contributors, along with their particular areas of expertise, as appropriate.

HISTORY

Document here the significant changes in each release of the product.

RESOURCES

If your product is designed to work under X windows, document here any X resources that affect the product's behavior.

FILES

Document here all files, or at least their directories if there are too many files. Also mention here any files in the user's home area that are needed/accessed (e.g., `$HOME/.mh_profile`, `$HOME/Mail/components` for the **mh** and **exmh** products).

BUGS

Document here things that do not (yet!) work as designed. Provide work-arounds whenever possible.

CAVEATS

Document here things that work *as designed* but which may be unclear or surprising to the user. (This is the System V replacement for the **BUGS** category; you too can pretend your product has no bugs!)

SEE ALSO

Document here other related commands and manual sections, especially if not obvious.

39.1.3 Example Source File

In section 17.1.5 we presented a simple example for the product **hello** showing how to create a formatted man page from a simple unformatted **nroff** input file. We will expand upon it here to illustrate the macros listed above. The **nroff** source is created in `$HELLO_DIR/man/hello.1`. Sample contents:

```
.TH HELLO 1 LOCAL
.SH NAME
hello - print "Hello world" on stdout
.SH SYNOPSIS
.B hello [options]
.I option option
.B ["
.I -yy -zz
.B ..."]
.SH AVAILABILITY
All UNIX flavors
.SH DESCRIPTION
hello prints the string "Hello world" on standard output.
.SH OPTIONS
There are no options, but we’ll make some up.
.TP 5
-yy
is one option
.TP
-zz
is another option
.SH AUTHOR
U. R. Friendly
```

39.2 Formatting the Source File

39.2.1 nroff

To create an ascii-formatted man page, you can run the utility **nroff** with the **`-man`** macro package as follows:

```
% nroff -man <input_file> > <output_file>
```

We recommend following the prescription for unformatted and formatted man page locations as stated above and in section 16.3. This ensures that the source file always gets run through the formatter and the formatted file is never run through it again, which would produce odd results. First, **`cd`** to the source file directory:

```
% cd $HELLO_DIR/man
```

The following command creates the formatted man page for our **hello** example in the correct directory:

```
% nroff -man hello.1 > ../catman/hello.1
```

Once it is formatted, the example above will look like this:

```
HELLO(1)                                                HELLO(1)



NAME
       hello - print "Hello world" on stdout

SYNOPSIS
       hello [options] option option [" -yy -zz ..."]

AVAILABILITY
       All UNIX flavors

DESCRIPTION
       hello  prints the string "Hello world" on standard output.

OPTIONS
       There are no options, but we’ll make some up.

       -yy  is one option
       -zz  is another option

AUTHOR
       U. R. Friendly

                              LOCAL                           1
```

39.2.2 groff

You can also use **groff** to format your man page source file. You must setup **groff** before use (not necessary for **nroff**). The command:

```
% groff -man -Tascii <input_file> > <ascii_output_file>
```

produces ascii-formatted man pages (the same output as the **nroff** command above). If you want to produce a PostScript output file, enter:

```
% groff -man <input_file> > <ps_output_file>
```

39.3 Converting your Man Page to html Format

An ascii-formatted man page can be run through the utility **man2html** and then accessed via a Web browser. First setup **conv2html**, then run the command:

```
% man2html -title ’<manpage_title>’ < <ascii_output_file> >\
  <html_file>
```