

NAME

groff_trace – groff macro package trace.tmac

SYNOPSIS

groff -m trace [*options...*] [*files...*]

Elements in brackets denote optional arguments, and the ellipsis means that there can be any number of arguments of this kind.

DESCRIPTION

The *trace* macro package of **groff**(1) can be a valuable tool for debugging documents written in the roff formatting language. A call stack trace is protocolled on standard error, that means, a diagnostic message is emitted on entering and exiting of a macro call. This greatly eases to track down an error in some macro.

This tracing process is activated by specifying the groff or troff command line option **-m trace**. This works also with the **groffer**(1) viewer program. A finer control can be obtained by including the macro file within the document by the groff macro call **.mso trace.tmac**. Only macros that are defined after this line are traced.

If some other macro package should be traced as well it must be specified after **-m trace** on the command line.

The macro file **trace.tmac** is unusual because it does not contain any macros to be called by a user. Instead, the existing macro definition and appending facilities are modified such that they display diagnostic messages.

EXAMPLES

In the following examples, a roff fragment is fed into groff via standard input. As we are only interested in the diagnostic messages (standard error) on the terminal, the normal formatted output (standard output) is redirected into the nirvana device */dev/null*. The resulting diagnostic messages are displayed directly below the corresponding example.

Command line option

```
sh# echo '.
> .de test_macro
> ..
> .test_macro
> .test_macro some dummy arguments
> ' | groff -m trace >/dev/null

*** de trace enter: test_macro
*** trace exit: test_macro
*** de trace enter: test_macro "some" "dummy" "arguments"
*** trace exit: test_macro "some" "dummy" "arguments"
```

The entry and the exit of each macro call is displayed on the terminal (standard output) — together with the arguments (if any).

Nested macro calls

```
sh# echo '.
> .de child
> ..
> .de parent
> .child
> ..
> .parent
> ' | groff -m trace >/dev/null

*** de trace enter: parent
*** de trace enter: child
*** trace exit: child
*** trace exit: parent
```

This shows that macro calls can be nested. This powerful feature can help to tack down quite complex call stacks.

Activating with .mso

```
sh# echo '.
> .de before
> ..
> .mso trace.tmac
> .de after
> ..
> .before
> .after
> .before
> ' | groff >/dev/null

*** de trace enter: after
*** trace exit: after
```

Here, the tracing is activated within the document, not by a command line option. As tracing was not active when macro *before* was defined, no call of this macro is protocolled; on the other hand, the macro *after* is fully protocolled.

FILES

The *trace* macros are kept in the file **trace.tmac** located in the *tmac* directory; see **groff_tmac(5)** for details.

ENVIRONMENT

\$GROFF_TMAC_PATH

A colon-separated list of additional tmac directories in which to search for macro files; see **groff_tmac(5)** for details.

AUTHOR

Copyright (C) 2002 Free Software Foundation, Inc.

This document is distributed under the terms of the FDL (GNU Free Documentation License) version 1.1 or later. You should have received a copy of the FDL on your system, it is also available on-line at the [GNU copyleft site](http://www.gnu.org/copyleft/fdl.html) (<http://www.gnu.org/copyleft/fdl.html>).

This document is part of *groff*, the GNU roff distribution. It was written by [Bernd Warken](mailto:bwarken@mayn.de) (bwarken@mayn.de).

SEE ALSO

groff(1)

An overview of the groff system.

troff(1) For details on option **-m**.

groffer(1)

A viewer program for all kinds of roff documents.

groff_tmac(5)

A general description of groff macro packages.

groff(7)

A short reference for the groff formatting language.

A complete reference for all parts of the groff system is found in the groff **info(1)** file.