

**TPLOG(3)**

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>SYNOPSIS</b>	<b>1</b>
<b>2</b>	<b>DESCRIPTION</b>	<b>2</b>
<b>3</b>	<b>RETURN VALUE</b>	<b>3</b>
<b>4</b>	<b>ERRORS</b>	<b>4</b>
<b>5</b>	<b>EXAMPLE</b>	<b>5</b>
<b>6</b>	<b>BUGS</b>	<b>6</b>
<b>7</b>	<b>SEE ALSO</b>	<b>7</b>
<b>8</b>	<b>COPYING</b>	<b>8</b>

## Chapter 1

# SYNOPSIS

```
#include <ndebug.h>
```

```
void tplog(int lev, char *message);
```

or

```
#define TP_LOG(lev, fmt, ...)
```

Link with *-lnstd -lpthread -lrt -lm*

## Chapter 2

# DESCRIPTION

Function writes the message to current log file. *lev* is logging level. For function version log message goes as *message* parameter. For C/C++ coding it is recommended to use macro version **TP\_LOG**, as it have a format string and it uses more efficient way to check the logging levels. Macro version have less functions calls during the whole process.

Log levels are following (from `ndebug.h`):

- `log_always` 1
  - `log_error` 2
  - `log_warn` 3
  - `log_info` 4
  - `log_debug` 5
-

## Chapter 3

# RETURN VALUE

N/A

## Chapter 4

# ERRORS

N/A

---

## Chapter 5

# EXAMPLE

Code logtest.c

```
#include <ndebug.h>

int main (int argc, char **argv)
{
    tplog(5, "Hello from function logger");

    TP_LOG(log_debug, "Hello from macro logger [logging level %d]", log_debug);

    return 0;
}
```

Sample run:

```
$ cc logtest.c -lnstd -lpthread -lrt -lm -I /include
$ ./a.out
t:USER:5:11064:000:20160928:100225252:/tplog.c:0412:Hello from function logger
t:USER:5:11064:000:20160928:100225252:ogtest.c:0007:Hello from macro logger [logging level 5] ←
```



## Chapter 6

# BUGS

Report bugs to [madars.vitolins@gmail.com](mailto:madars.vitolins@gmail.com)

## Chapter 7

## SEE ALSO

**tplogdump(3) tplogdumpdiff(3) tplogprintubf(3) tplogconfig(3) tplogsetreqfile\_direct(3) tplogsetreqfile(3) ex\_devguide(guides)**

## Chapter 8

# COPYING

© Mavimax, Ltd