



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

***Rocky Enterprise Linux 9.2 Manual Pages on command 'Apache::TestTrace.3pm'***

***\$ man Apache::TestTrace.3pm***

Apache::TestTrace(3pm)      User Contributed Perl Documentation      Apache::TestTrace(3pm)

NAME

Apache::TestTrace - Helper output generation functions

SYNOPSIS

```
use Apache::TestTrace;

debug "foo bar";

info_sub "missed it";

error_mark "something is wrong";

# test sub that exercises all the tracing functions

sub test {

    print $Apache::TestTrace::LogFH

        "TraceLevel: $Apache::TestTrace::Level\n";

    $_->($_,[1..3],$_) for qw(emerg alert crit error

        warning notice info debug todo);

    print $Apache::TestTrace::LogFH "\n\n"

};

# demo the trace subs using default setting

test();

{

    # override the default trace level with 'crit'

    local $Apache::TestTrace::Level = 'crit';

    # now only 'crit' and higher levels will do tracing lower level

    test();
```

```

}

{
    # set the trace level to 'debug'
    local $Apache::TestTrace::Level = 'debug';
    # now only 'debug' and higher levels will do tracing lower level
    test();
}

{
    open OUT, ">/tmp/foo" or die $!;
    # override the default Log filehandle
    local $Apache::TestTrace::LogFH = \*OUT;
    # now the traces will go into a new filehandle
    test();
    close OUT;
}

# override tracing level via -trace opt
% t/TEST -trace=debug

# override tracing level via env var
% env APACHE_TEST_TRACE_LEVEL=debug t/TEST

```

## DESCRIPTION

This module exports a number of functions that make it easier generating various diagnostics messages in your programs in a consistent way and saves some keystrokes as it handles the new lines and sends the messages to STDERR for you.

This module provides the same trace methods as syslog(3)'s log levels. Listed from low level to high level: `emerg()`, `alert()`, `crit()`, `error()`, `warning()`, `notice()`, `info()`, `debug()`. The only different function is `warning()`, since `warn` is already taken by Perl.

The module provides another trace function called `todo()` which is useful for todo items. It has the same level as `debug` (the highest).

There are two more variants of each of these functions. If the `_mark` suffix is appended (e.g., `error_mark`) the trace will start with the filename and the line number the function was called from. If the `_sub` suffix is appended (e.g., `error_info`) the trace will start with the name of the subroutine the function was called from.

If you have "Term::ANSIColor" installed the diagnostic messages will be colored,

otherwise a special for each function prefix will be used.

If "Data::Dumper" is installed and you pass a reference to a variable to any of these functions, the variable will be dumped with "Data::Dumper::Dumper()".

Functions whose level is above the level set in `$Apache::TestTrace::Level` become NOPs. For example if the level is set to `alert`, only `alert()` and `emerg()` functions will generate the output. The default setting of this variable is `warning`. Other valid values are: `emerg`, `alert`, `crit`, `error`, `warning`, `notice`, `info`, `debug`.

Another way to affect the trace level is to set `$ENV{APACHE_TEST_TRACE_LEVEL}`, which takes effect if `$Apache::TestTrace::Level` is not set. So an explicit setting of `$Apache::TestTrace::Level` always takes precedence.

By default all the output generated by these functions goes to `STDERR`. You can override the default filehandler by overriding `$Apache::TestTrace::LogFH` with a new filehandler.

When you override this package's global variables, think about localizing your local settings, so it won't affect other modules using this module in the same run.

#### TODO

- o provide an option to disable the coloring altogether via some flag
- o `import()`

#### AUTHOR

Stas Bekman with contributions from Doug MacEachern

perl v5.34.0

2022-02-06

Apache::TestTrace(3pm)