



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 'Test2::Util.3perl'

\$ man Test2::Util.3perl

Test2::Util(3perl) Perl Programmers Reference Guide Test2::Util(3perl)

NAME

Test2::Util - Tools used by Test2 and friends.

DESCRIPTION

Collection of tools used by Test2 and friends.

EXPORTS

All exports are optional. You must specify subs to import.

```
($success, $error) = try { ... }
```

Eval the codeblock, return success or failure, and the error message. This code protects \$@ and \$!, they will be restored by the end of the run. This code also temporarily blocks \$SIG{DIE} handlers.

```
protect { ... }
```

Similar to try, except that it does not catch exceptions. The idea here is to protect \$@ and \$! from changes. \$@ and \$! will be restored to whatever they were before the run so long as it is successful. If the run fails \$! will still be restored, but \$@ will contain the exception being thrown.

True if this system is capable of true or pseudo-fork.

CAN_REALLY_FORK

True if the system can really fork. This will be false for systems where fork is emulated.

CAN_THREAD

True if this system is capable of using threads.

USE_THREADS

Returns true if threads are enabled, false if they are not.

get_tid

This will return the id of the current thread when threads are enabled, otherwise it returns 0.

my \$file = pkg_to_file(\$package)

Convert a package name to a filename.

\$string = ipc_separator()

Get the IPC separator. Currently this is always the string '~'.

\$string = gen_uid()

Generate a unique id (NOT A UUID). This will typically be the process id, the thread id, the time, and an incrementing integer all joined with the "ipc_separator()".

These ID's are unique enough for most purposes. For identical ids to be generated you must have 2 processes with the same PID generate IDs at the same time with the same current state of the incrementing integer. This is a perfectly reasonable thing to expect to happen across multiple machines, but is quite unlikely to happen on one machine.

This can fail to be unique if a process generates an id, calls exec, and does it again

after the exec and it all happens in less than a second. It can also happen if the systems process id's cycle in less than a second allowing 2 different programs that use this generator to run with the same PID in less than a second. Both these cases are sufficiently unlikely. If you need universally unique ids, or ids that are unique in these conditions, look at Data::UUID.

```
($ok, $err) = do_rename($old_name, $new_name)
```

Rename a file, this wraps "rename()" in a way that makes it more reliable cross-platform when trying to rename files you recently altered.

```
($ok, $err) = do_unlink($filename)
```

Unlink a file, this wraps "unlink()" in a way that makes it more reliable cross-platform when trying to unlink files you recently altered.

```
($ok, $err) = try_sig_mask { ... }
```

Complete an action with several signals masked, they will be unmasked at the end allowing any signals that were intercepted to get handled.

This is primarily used when you need to make several actions atomic (against some signals anyway).

Signals that are intercepted:

SIGINT

SIGALRM

SIGHUP

SIGTERM

SIGUSR1

SIGUSR2

NOTES && CAVEATS

5.10.0

Perl 5.10.0 has a bug when compiled with newer gcc versions. This bug causes a

segfault whenever a new thread is launched. Test2 will attempt to detect this, and note that the system is not capable of forking when it is detected.

Devel::Cover

Devel::Cover does not support threads. CAN_THREAD will return false if Devel::Cover is loaded before the check is first run.

SOURCE

The source code repository for Test2 can be found at <http://github.com/Test-More/test-more/>.

MAINTAINERS

Chad Granum <exodist@cpan.org>

AUTHORS

Chad Granum <exodist@cpan.org>

Kent Fredric <kentnl@cpan.org>

COPYRIGHT

Copyright 2020 Chad Granum <exodist@cpan.org>.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

See <http://dev.perl.org/licenses/>

perl v5.34.0

2023-11-23

Test2::Util(3perl)