



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 'Sub::Util.3perl'

\$ man Sub::Util.3perl

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

NAME

Sub::Util - A selection of utility subroutines for subs and CODE references

SYNOPSIS

```
use Sub::Util qw( prototype set_prototype subname set_subname );
```

DESCRIPTION

"Sub::Util" contains a selection of utility subroutines that are useful for operating on subs and CODE references.

The rationale for inclusion in this module is that the function performs some work for which an XS implementation is essential because it cannot be implemented in Pure Perl, and which is sufficiently-widely used across CPAN that its popularity warrants inclusion in a core module, which this is.

FUNCTIONS

prototype

```
my $proto = prototype( $code )
```

Since version 1.40.

Returns the prototype of the given \$code reference, if it has one, as a string. This is the same as the "CORE::prototype" operator; it is included here simply for symmetry and completeness with the other functions.

set_prototype

```
my $code = set_prototype $prototype, $code;
```

Since version 1.40.

Sets the prototype of the function given by the \$code reference, or deletes it if \$prototype is "undef". Returns the \$code reference itself.

Caution: This function takes arguments in a different order to the previous copy of the code from "Scalar::Util". This is to match the order of "set_subname", and other potential additions in this file. This order has been chosen as it allows a neat and simple chaining of other "Sub::Util::set_*" functions as might become available, such as:

```
my $code =  
  set_subname name_here =>  
  set_prototype '&@' =>  
  set_attribute ':lvalue' =>  
  sub { ..... };
```

subname

```
my $name = subname( $code )
```

Since version 1.40.

Returns the name of the given \$code reference, if it has one. Normal named subs will give a fully-qualified name consisting of the package and the localname separated by "::".

Anonymous code references will give "__ANON__" as the localname. If the package the code was compiled in has been deleted (e.g. using "delete_package" from Symbol), "__ANON__" will be returned as the package name. If a name has been set using "set_subname", this

name will be returned instead.

This function was inspired by "sub_fullname" from Sub::Identify. The remaining functions that "Sub::Identify" implements can easily be emulated using regexp operations, such as

```
sub get_code_info { return (subname $_[0]) =~ m/^(.+):(.*?)$/ }
sub sub_name     { return (get_code_info $_[0])[0] }
sub stash_name   { return (get_code_info $_[0])[1] }
```

Users of Sub::Name beware: This function is not the same as "Sub::Name::subname"; it returns the existing name of the sub rather than changing it. To set or change a name, see instead "set_subname".

set_subname

```
my $code = set_subname $name, $code;
```

Since version 1.40.

Sets the name of the function given by the \$code reference. Returns the \$code reference itself. If the \$name is unqualified, the package of the caller is used to qualify it.

This is useful for applying names to anonymous CODE references so that stack traces and similar situations, to give a useful name rather than having the default of "__ANON__".

Note that this name is only used for this situation; the "set_subname" will not install it into the symbol table; you will have to do that yourself if required.

However, since the name is not used by perl except as the return value of "caller", for stack traces or similar, there is no actual requirement that the name be syntactically valid as a perl function name. This could be used to attach extra information that could be useful in debugging stack traces.

This function was copied from "Sub::Name::subname" and renamed to the naming convention of this module.

AUTHOR

The general structure of this module was written by Paul Evans <leonerd@leonerd.org.uk>.

The XS implementation of "set_subname" was copied from Sub::Name by Matthijs van Duin <xmath@cpan.org>

perl v5.34.0

2023-11-23

Sub::Util(3perl)