



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 'ModPerl::Code.3pm'

\$ man ModPerl::Code.3pm

ModPerl::Code(3pm) User Contributed Perl Documentation ModPerl::Code(3pm)

NAME

`$class\::Const` - Perl Interface for `$class` Constants

SYNOPSIS

CONSTANTS

EOF

```
my $groups = $data{$class};  
for my $group (sort keys %$groups) {  
    print $fh <<"EOF";
```

`":$group"`

```
use $class\::Const -compile qw(:$group);
```

The `":$group"` group is for XXX constants.

EOF

```
for my $const (sort @{$groups->{$group}}) {  
    print $fh "=head3 C<$class\::$const>\n\n\n";  
}
```

```

    }

    print $fh "=cut\n";
}
}

sub generate_constants_lookup_doc {
    my ($data) = @_;

    foreach my $class (sort keys %$Apache2::ConstantsTable) {
        my $groups = $Apache2::ConstantsTable->{$class};
        my $constants = [sort map { @$_ } values %$groups];

        constants_lookup_code_doc($constants, $class, $data);
    }
}

sub generate_constants_group_lookup_doc {
    my ($data) = @_;

    foreach my $class (sort keys %$Apache2::ConstantsTable) {
        my $groups = $Apache2::ConstantsTable->{$class};
        constants_group_lookup_code_doc($class, $groups, $data);
    }
}

sub constants_group_lookup_code_doc {
    my ($class, $groups, $data) = @_;

    my @tags;
    my @code;

    while (my ($group, $constants) = each %$groups) {
        $data->{$class}{$group} = [

```

```

map {
    my @ifdef = constants_ifdef($_);
    s/^(constant_prefixes)_?//o;
    $seen_const{$class}{$_}++;
    $_;
} @$constants
];
}
}

sub constants_lookup_code_doc {
    my ($constants, $class, $data) = @_;

    my (%switch, %alias);

    %alias = %shortcuts;

    my $postfix = lc $class;
    my $package = $class . '::';
    my $package_len = length $package;

    my $func = canon_func(qw(constants lookup), $postfix);

    for (@$constants) {
        if (s/^(constant_prefixes)(_)?//o) {
            $alias{$_} = join $2 || "", $1, $_;
        }
        else {
            $alias{$_} ||= $_;
        }
        next unless /^[A-Z]/;
        push @{$switch{$1}}, $_;
    }
}

```

```

for my $key (sort keys %switch) {
    my $names = $switch{$key};
    for my $name (@$names) {
        my @ifdef = constants_ifdef($alias{$name});
        push @{$data->{$class}{other} }, $name
            unless $seen_const{$class}{$name}
    }
}
}
}

```

```

sub generate_exports {
    my ($self, $c_fh) = @_;
    require ModPerl::WrapXS;
    ModPerl::WrapXS->generate_exports($c_fh); }

```

src/modules/perl/*.c files needed to build APR/APR::* outside # of mod_perl.so sub

```

src_apr_ext {
    return map { "modperl_$_" } (qw(error bucket),
        map { "common_$_" } qw(util log)); }

```

1; __END__

NAME

ModPerl::Code - Generate mod_perl glue code

SYNOPSIS

```

use ModPerl::Code ();

my $code = ModPerl::Code->new;

$code->generate;

```

DESCRIPTION

This module provides functionality for generating mod_perl glue code. Reason this code is

generated rather than written by hand include:

consistency

thin and clean glue code

enable/disable features (without #ifdefs)

adapt to changes in Apache

experiment with different approaches to gluing

AUTHOR

Doug MacEachern

perl v5.34.0

2022-02-06

ModPerl::Code(3pm)