

# **1 Apache::Symbol - Things for symbol things**

## 1.1 Synopsis

```
use Apache::Symbol ();

@ISA = qw(Apache::Symbol);
```

## 1.2 Description

`Apache::Symbol` helps `mod_perl` users avoid Perl warnings related with redefined constant functions.

`perlsub/Constant Functions` says:

```
If you redefine a subroutine which was eligible for inlining you'll
get a mandatory warning. (You can use this warning to tell whether
or not a particular subroutine is considered constant.) The warning
is considered severe enough not to be optional because previously
compiled invocations of the function will still be using the old
value of the function.
```

*mandatory warning* means there is **no** way to avoid this warning no matter what tricks you pull in Perl. This is bogus for us `mod_perl` users when restarting the server with `PerlFreshRestart` on or when `Apache::StatINC` pulls in a module that has changed on disk.

You can, however, pull some tricks with XS to avoid this warning, `Apache::Symbol::undef_functions` does just that.

## 1.3 Arguments

`undef_functions` takes two arguments: `skip` and `only_undef_exports`.

`skip` is a regular expression indicating the function names to skip.

Use the `only_undef_exports` flag to undef only those functions which are listed in `@EXPORT`, `@EXPORT_OK`, `%EXPORT_TAGS`, or `@EXPORT_EXTRAS`. `@EXPORT_EXTRAS` is not used by the Exporter, it only exists to communicate with `undef_functions`.

As a special case, if none of the `EXPORT` variables are defined ignore `only_undef_exports`. This takes care of trivial modules that don't use the Exporter.

## 1.4 Players

This module and the undefining of functions is optional, if you wish to have this functionality enabled, there are one or more switches you need to know about.

- **PerlRestartHandler**

Apache::Symbol defines a `PerlRestartHandler` which can be useful in conjunction with `PerlFreshRestart` On as it will avoid subroutine redefinition messages. Configure like so:

```
PerlRestartHandler Apache::Symbol
```

- **Apache::Registry**

By placing the *SYNOPSIS* bit in you script, `Apache::Registry` will undefine subroutines in your script before it is re-compiled to avoid "subroutine re-defined" warnings.

- **Apache::StatINC**

See `Apache::StatINC`'s docs.

- **APACHE\_SYMBOL\_UNIVERSAL**

If this environment variable is true when `Symbol.pm` is compiled, it will define `UNIVERSAL::undef_functions`, which means all classes will inherit **Apache::Symbol::undef\_functions**.

- **Others**

Modules such as `HTML::Embperl` and `Apache::ePerl` which compile and script cache scripts ala `Apache::Registry` style can use `undef_functions` with this bit of code:

```
if($package->can('undef_functions')) {
    $package->undef_functions;
}
```

Where `$package` is the name of the package in which the script is being re-compiled.

## 1.5 See Also

`perlsub`, `Devel::Symdump`

## 1.6 Maintainers

Maintainer is the person(s) you should contact with updates, corrections and patches.

- **The documentation mailing list**

## 1.7 Authors

- **Doug MacEachern**

Only the major authors are listed above. For contributors see the Changes file.



## Table of Contents:

1	Apache::Symbol - Things for symbol things	1
1.1	Synopsis	2
1.2	Description	2
1.3	Arguments	2
1.4	Players	2
1.5	See Also	3
1.6	Maintainers	3
1.7	Authors	3