NAME

warnings - Perl pragma to control optional warnings

SYNOPSIS

```
use warnings;
no warnings;

use warnings "all";
no warnings "all";

use warnings::register;
if (warnings::enabled()) {
    warnings::warn("some warning");
}

if (warnings::enabled("void")) {
    warnings::warn("void", "some warning");
}

if (warnings::enabled($object)) {
    warnings::warn($object, "some warning");
}

warnings::warnif("some warning");
warnings::warnif("void", "some warning");
warnings::warnif($object, "some warning");
```

DESCRIPTION

The `warnings` pragma is a replacement for the command line flag `-w`, but the pragma is limited to the enclosing block, while the flag is global. See *perllexwarn* for more information and the list of built-in warning categories.

If no import list is supplied, all possible warnings are either enabled or disabled.

A number of functions are provided to assist module authors.

use warnings::register

Creates a new warnings category with the same name as the package where the call to the pragma is used.

warnings::enabled()

Use the warnings category with the same name as the current package.

Return TRUE if that warnings category is enabled in the calling module. Otherwise returns FALSE.

warnings::enabled($category)

Return TRUE if the warnings category, `$category`, is enabled in the calling module. Otherwise returns FALSE.

warnings::enabled($object)

Use the name of the class for the object reference, `$object`, as the warnings category.

Return TRUE if that warnings category is enabled in the first scope where the object is used. Otherwise returns FALSE.

warnings::fatal_enabled()

Return TRUE if the warnings category with the same name as the current package has been set to FATAL in the calling module. Otherwise returns FALSE.

warnings::fatal_enabled($category)

Return TRUE if the warnings category `$category` has been set to FATAL in the calling module. Otherwise returns FALSE.

warnings::fatal_enabled($object)

Use the name of the class for the object reference, `$object`, as the warnings category.

Return TRUE if that warnings category has been set to FATAL in the first scope where the object is used. Otherwise returns FALSE.

warnings::warn($message)

Print `$message` to STDERR.

Use the warnings category with the same name as the current package.

If that warnings category has been set to "FATAL" in the calling module then die. Otherwise return.

warnings::warn($category, $message)

Print `$message` to STDERR.

If the warnings category, `$category`, has been set to "FATAL" in the calling module then die. Otherwise return.

warnings::warn($object, $message)

Print `$message` to STDERR.

Use the name of the class for the object reference, `$object`, as the warnings category.

If that warnings category has been set to "FATAL" in the scope where `$object` is first used then die. Otherwise return.

warnings::warnif($message)

Equivalent to:

```
if (warnings::enabled())
  { warnings::warn($message) }
```

warnings::warnif($category, $message)

Equivalent to:

```
if (warnings::enabled($category))
  { warnings::warn($category, $message) }
```

warnings::warnif($object, $message)

Equivalent to:

```
if (warnings::enabled($object))
  { warnings::warn($object, $message) }
```

warnings::register_categories(@names)

This registers warning categories for the given names and is primarily for use by the warnings::register pragma, for which see *perllexwarn*.

See *"Pragmatic Modules" in perlmodlib* and *perllexwarn*.