

Empty %s

(F) \p and \P are used to introduce a named Unicode property, as described in *perlunicode* and *perlre*. You used \p or \P in a regular expression without specifying the property name.

entering effective %s failed

(F) While under the use `filetest` pragma, switching the real and effective uids or gids failed.

Error converting file specification %s

(F) An error peculiar to VMS. Because Perl may have to deal with file specifications in either VMS or Unix syntax, it converts them to a single form when it must operate on them directly. Either you've passed an invalid file specification to Perl, or you've found a case the conversion routines don't handle. Drat.

%s: Eval-group in insecure regular expression

(F) Perl detected tainted data when trying to compile a regular expression that contains the `(?{ ... })` zero-width assertion, which is unsafe. See `(?{ code })` in *perlre*, and *perlsec*.

%s: Eval-group not allowed at run time

(F) Perl tried to compile a regular expression containing the `(?{ ... })` zero-width assertion at run time, as it would when the pattern contains interpolated values. Since that is a security risk, it is not allowed. If you insist, you may still do this by explicitly building the pattern from an interpolated string at run time and using that in an `eval()`. See `(?{ code })` in *perlre*.

%s: Eval-group not allowed, use re 'eval'

(F) A regular expression contained the `(?{ ... })` zero-width assertion, but that construct is only allowed when the use `re 'eval'` pragma is in effect. See `(?{ code })` in *perlre*.

Excessively long <> operator

(F) The contents of a `<>` operator may not exceed the maximum size of a Perl identifier. If you're just trying to glob a long list of filenames, try using the `glob()` operator, or put the filenames into a variable and glob that.

exec? I'm not *that* kind of operating system

(F) The `exec` function is not implemented in MacPerl. See *perlport*.

Execution of %s aborted due to compilation errors

(F) The final summary message when a Perl compilation fails.

Exiting eval via %s

(W exiting) You are exiting an eval by unconventional means, such as a `goto`, or a loop control statement.

Exiting format via %s

(W exiting) You are exiting a format by unconventional means, such as a `goto`, or a loop control statement.

Exiting pseudo-block via %s

(W exiting) You are exiting a rather special block construct (like a sort block or subroutine) by unconventional means, such as a `goto`, or a loop control statement. See `sort` in *perlfunc*.

Exiting subroutine via %s

(W exiting) You are exiting a subroutine by unconventional means, such as a `goto`, or a loop control statement.

Exiting substitution via %s

(W exiting) You are exiting a substitution by unconventional means, such as a `return`, a `goto`, or a loop control statement.

Explicit blessing to '' (assuming package main)

(W misc) You are blessing a reference to a zero length string. This has the effect of blessing the reference into the package `main`. This is usually not what you want. Consider providing a default target package, e.g. `bless($ref, $p || 'MyPackage');`