

**NAME**

Pod::Find - find POD documents in directory trees

**SYNOPSIS**

```
use Pod::Find qw(pod_find simplify_name);
my %pods = pod_find({ -verbose => 1, -inc => 1 });
foreach(keys %pods) {
    print "found library POD `$_' in $_\n";
}

print "podname=",simplify_name('a/b/c/mymodule.pod'),"\n";

$location = pod_where( { -inc => 1 }, "Pod::Find" );
```

**DESCRIPTION**

**NOTE: This module is considered legacy; modern Perl releases (5.18 and higher) are going to remove Pod-Parser from core and use Pod-Simple for all things POD.**

**Pod::Find** provides a set of functions to locate POD files. Note that no function is exported by default to avoid pollution of your namespace, so be sure to specify them in the **use** statement if you need them:

```
use Pod::Find qw(pod_find);
```

From this version on the typical SCM (software configuration management) directories are ignored. These are: RCS, CVS, SCCS, .svn, .hg, .git, .sync

```
pod_find( { %opts } , @directories )
```

The function **pod\_find** searches for POD documents in a given set of files and/or directories. It returns a hash with the file names as keys and the POD name as value. The POD name is derived from the file name and its position in the directory tree.

E.g. when searching in *\$HOME/perl5lib*, the file *\$HOME/perl5lib/MyModule.pm* would get the POD name *MyModule*, whereas *\$HOME/perl5lib/Myclass/Subclass.pm* would be *Myclass::Subclass*. The name information can be used for POD translators.

Only text files containing at least one valid POD command are found.

A warning is printed if more than one POD file with the same POD name is found, e.g. *CPAN.pm* in different directories. This usually indicates duplicate occurrences of modules in the *@INC* search path.

**OPTIONS** The first argument for **pod\_find** may be a hash reference with options. The rest are either directories that are searched recursively or files. The POD names of files are the plain basenames with any Perl-like extension (.pm, .pl, .pod) stripped.

```
-verbose => 1
```

Print progress information while scanning.

```
-perl => 1
```

Apply Perl-specific heuristics to find the correct PODs. This includes stripping Perl-like extensions, omitting subdirectories that are numeric but do *not* match the current Perl interpreter's version id, suppressing *site\_perl* as a module hierarchy name etc.

```
-script => 1
```

Search for PODs in the current Perl interpreter's installation **scriptdir**. This is taken from the local Config module.

```
-inc => 1
```

Search for PODs in the current Perl interpreter's *@INC* paths. This automatically considers paths specified in the *PERL5LIB* environment as this is included in *@INC* by the Perl interpreter itself.

```
simplify_name( $str )
```

The function **simplify\_name** is equivalent to **basename**, but also strips Perl-like extensions (.pm, .pl, .pod) and extensions like *.bat*, *.cmd* on Win32 and OS/2, or *.com* on VMS, respectively.

```
pod_where( { %opts }, $pod )
```

Returns the location of a pod document given a search directory and a module (e.g. `File::Find` or script (e.g. `perldoc(1)` name).

Options:

```
-inc => 1
```

Search `@INC` for the pod and also the `scriptdir` defined in the `Config` module.

```
-dirs => [ $dir1, $dir2, ... ]
```

Reference to an array of search directories. These are searched in order before looking in `@INC` (if **-inc**). Current directory is used if none are specified.

```
-verbose => 1
```

List directories as they are searched

Returns the full path of the first occurrence to the file. Package names (eg `'A::B'`) are automatically converted to directory names in the selected directory. (eg on unix `'A::B'` is converted to `'A/B'`). Additionally, `'.pm'`, `'.pl'` and `'.pod'` are appended to the search automatically if required.

A subdirectory `pod/` is also checked if it exists in any of the given search directories. This ensures that e.g. `perlfunc(1)` is found.

It is assumed that if a module name is supplied, that that name matches the file name. Pods are not opened to check for the `'NAME'` entry.

A check is made to make sure that the file that is found does contain some pod documentation.

```
contains_pod( $file , $verbose )
```

Returns true if the supplied filename (not POD module) contains some pod information.

## AUTHOR

Please report bugs using <http://rt.cpan.org>.

Marek Rouchal <marekr@cpan.org>, heavily borrowing code from Nick Ing-Simmons' `PodToHtml`.

Tim Jenness <t.jenness@jach.hawaii.edu> provided `pod_where` and `contains_pod`.

**Pod::Find** is part of the `Pod::Parser` distribution.

## SEE ALSO

[Pod::Parser](#), [Pod::Checker](#), [perldoc\(1\)](#)