

**NAME**

CPAN::Distroprefs -- read and match distroprefs

**SYNOPSIS**

```
use CPAN::Distroprefs;

my %info = (... distribution/environment info ...);

my $finder = CPAN::Distroprefs->find($prefs_dir, \%ext_map);

while (my $result = $finder->next) {

    die $result->as_string if $result->is_fatal;

    warn($result->as_string), next if $result->is_warning;

    for my $pref (@{ $result->prefs }) {
        if ($pref->matches(\%info)) {
            return $pref;
        }
    }
}
```

**DESCRIPTION**

This module encapsulates reading Distroprefs and matching them against CPAN distributions.

**INTERFACE**

```
my $finder = CPAN::Distroprefs->find($dir, \%ext_map);

while (my $result = $finder->next) { ... }
```

Build an iterator which finds distroprefs files in the tree below the given directory. Within the tree directories matching `m/^[. _]/` are pruned.

`%ext_map` is a hashref whose keys are file extensions and whose values are modules used to load matching files:

```
{
    'yaml' => 'YAML::Syck',
    'dd' => 'Data::Dumper',
    ...
}
```

Each time `$finder->next` is called, the iterator returns one of two possible values:

- a `CPAN::Distroprefs::Result` object
- `undef`, indicating that no prefs files remain to be found

**RESULTS**

`find()` returns `CPAN::Distroprefs::Result` objects to indicate success or failure when reading a prefs file.

**Common**

All results share some common attributes:

*type*

success, warning, or fatal

*file*

the file from which these prefs were read, or to which this error refers (relative filename)

*ext*

the file's extension, which determines how to load it

*dir*

the directory the file was read from

*abs*

the absolute path to the file

### Errors

Error results (warning and fatal) contain:

*msg*

the error message (usually either \$! or a YAML error)

### Successes

Success results contain:

*prefs*

an arrayref of CPAN::Distroprefs::Pref objects

### PREFS

CPAN::Distroprefs::Pref objects represent individual distroprefs documents. They are constructed automatically as part of `success` results from `find()`.

*data*

the pref information as a hashref, suitable for e.g. passing to Kwalify

*match\_attributes*

returns a list of the valid match attributes (see the Distroprefs section in CPAN)

currently: `env perl perlconfig distribution module`

*has\_any\_match*

true if this pref has a 'match' attribute at all

*has\_valid\_subkeys*

true if this pref has a 'match' attribute and at least one valid match attribute

*matches*

```
if ($pref->matches(\%arg)) { ... }
```

true if this pref matches the passed-in hashref, which must have a value for each of the `match_attributes` (above)

### LICENSE

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.