

**NAME**

File::Find::Rule::Extending – the mini-guide to extending File::Find::Rule

**SYNOPSIS**

```
package File::Find::Rule::Random;
use strict;

# take useful things from File::Find::Rule
use base 'File::Find::Rule';

# and force our crack into the main namespace
sub File::Find::Rule::random () {
    my $self = shift()->_force_object;
    $self->exec( sub { rand > 0.5 } );
}

1;
```

**DESCRIPTION**

File::Find::Rule went down so well with the buying public that everyone wanted to add extra features. With the 0.07 release this became a possibility, using the following conventions.

**Declare your package**

```
package File::Find::Rule::Random;
use strict;
```

**Inherit methods from File::Find::Rule**

```
# take useful things from File::Find::Rule
use base 'File::Find::Rule';
```

*Force your madness into the main package*

```
# and force our crack into the main namespace
sub File::Find::Rule::random () {
    my $self = shift()->_force_object;
    $self->exec( sub { rand > 0.5 } );
}
```

Yes, we're being very cavalier here and defining things into the main File::Find::Rule namespace. This is due to lack of imagination on my part – I simply can't find a way for the functional and oo interface to work without doing this or some kind of inheritance, and inheritance stops you using two File::Find::Rule::Foo modules together.

For this reason try and pick distinct names for your extensions. If this becomes a problem then I may institute a semi-official registry of taken names.

**Taking no arguments.**

Note the null prototype on random. This is a cheat for the procedural interface to know that your sub takes no arguments, and so allows this to happen:

```
find( random => in => '.' );
```

If you hadn't declared random with a null prototype it would have consumed in as a parameter to it, then got all confused as it doesn't know about a ' .' rule.

**AUTHOR**

Richard Clamp <richardc@unixbeard.net>

**COPYRIGHT**

Copyright (C) 2002 Richard Clamp. All Rights Reserved.

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

## **SEE ALSO**

File::Find::Rule

File::Find::Rule::MMagic was the first extension module, so maybe check that out.