

NAME

Module::Install::PAR – Module::Install Support for PAR::Dist packages

SYNOPSIS

To offer your users the possibility to install binaries if we cannot compile an XS version of the module, you could use this simplistic stub:

```
use inc::Module::Install;

name          'Foo';
all_from      'lib/Foo.pm';

# Which CPAN directory do we fetch binaries from?
par_base      'SMUELLER';

unless ( can_xs ) {
    my $okay = extract_par( fetch_par );
    if ( not $okay ) {
        die "No compiler and no binary package found. Aborting.\n";
    }
}

WriteAll;
```

DESCRIPTION

This module adds a couple of directives to Module::Install related to installing and creating PAR::Dist distributions.

par_base

This directive sets the CPAN ID from whose CPAN directory to fetch binaries from. For example, you can choose to download binaries from <http://www.cpan.org/authors/id/S/SM/SMUELLER/> or its ftp counterpart by writing:

```
par_base 'SMUELLER';
```

By default, the name of the file to fetch is generated from the distribution name, its version, your platform name and your perl version concatenated with dashes.

The directive, however, takes an optional second argument which specifies the name of the file to fetch. (Though `par_base` does not fetch files itself, see below.)

```
par_base 'SMUELLER', 'foo';
```

Once `fetch_par` is called, the file 'foo' will be downloaded from SMUELLER's CPAN directory. (It doesn't exist.)

The second argument could be used to fetch platform-agnostic binaries:

```
par_base 'SMUELLER', "Some-Distribution-0.01.par";
```

(Documentation TODO: Use the previously defined distribution name and version in example.)

fetch_par

Fetches the .par file previously referenced in the documentation of the `par_base` directive.

`fetch_par` can be used without arguments given the `par_base` directive was used before. It will return the name of the file it fetched.

If the first argument is an URL or a CPAN user ID, the file is fetched from that directory unless an URL has been previously set. (Read that again.)

If the second argument is a file name it is used as the name of the file to download.

If the file could not be fetched, a suitable error message about no package being available, yada yada yada, is printed. You can turn this off by specifying a true third argument.



```
# Try to fetch the package (see par_base) but
# don't be verbose about failures
my $file = fetch_par('', '', undef);
```

extract_par

Takes the name of a PAR::Dist archive file as first argument. The 'blib/' directory of this archive is extracted and the 'pm_to_blib' is created.

Typical shorthand usage:

```
extract_par( fetch_par ) or die "Could not install PAR::Dist archive.";
```

make_par

This directive requires PAR::Dist (version 0.03 or up) on your system. (And checks that it is available before continuing.)

Creates a PAR::Dist archive from the 'blib/' subdirectory.

First argument must be the name of the PAR::Dist archive to create.

If your Makefile.PL has a `par_base` directive, the `make par` make target will be available. It uses this `make_par` directive internally, so on your development system, you can do this to create a .par binary archive for your platform:

```
perl Makefile.PL
make
make par
```

AUTHOR

Audrey Tang <cpan AT audreyt DOT org>

With documentation from Steffen Mueller <smueller AT cpan DOT org>

COPYRIGHT

Copyright (c) 2006. Audrey Tang.

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

See <<http://www.perl.com/perl/misc/Artistic.html>>

