



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'ExtUtils::Install.3perl'

\$ man ExtUtils::Install.3perl

ExtUtils::Install(3perl) Perl Programmers Reference Guide ExtUtils::Install(3perl)

NAME

ExtUtils::Install - install files from here to there

SYNOPSIS

```
use ExtUtils::Install;

install({ 'blib/lib' => 'some/install/dir' });

uninstall($packlist);

pm_to_blib({ 'lib/Foo/Bar.pm' => 'blib/lib/Foo/Bar.pm' });
```

VERSION

2.20

DESCRIPTION

Handles the installing and uninstalling of perl modules, scripts, man pages, etc...

Both install() and uninstall() are specific to the way ExtUtils::MakeMaker handles the installation and deinstallation of perl modules. They are not designed as general purpose tools.

On some operating systems such as Win32 installation may not be possible until after a reboot has occurred. This can have varying consequences: removing an old DLL does not impact programs using the new one, but if a new DLL cannot be installed properly until reboot then anything depending on it must wait. The package variable

`$ExtUtils::Install::MUST_REBOOT`

is used to store this status.

If this variable is true then such an operation has occurred and anything depending on this module cannot proceed until a reboot has occurred.

If this value is defined but false then such an operation has occurred, but should not impact later operations.

Functions

install

deprecated forms

```
install(\%from_to);
```

```
install(\%from_to, $verbose, $dry_run, $uninstall_shadows,  
        $skip, $always_copy, \%result);
```

recommended form as of 1.47

```
install(  
    from_to => \%from_to,  
    verbose => 1,  
    dry_run => 0,  
    uninstall_shadows => 1,  
    skip => undef,  
    always_copy => 1,  
    result => \%install_results,  
);
```

Copies each directory tree of %from_to to its corresponding value preserving timestamps and permissions.

There are two keys with a special meaning in the hash: "read" and "write". These contain packlist files. After the copying is done, install() will write the list of target files to \$from_to{write}. If \$from_to{read} is given the contents of this file will be merged into the written file. The read and the written file may be identical, but on AFS it is quite likely that people are installing to a different directory than the one where the files later appear.

If \$verbose is true, will print out each file removed. Default is false. This is "make install VERBINST=1". \$verbose values going up to 5 show increasingly more diagnostics output.

If \$dry_run is true it will only print what it was going to do without actually doing it.

Default is false.

If \$uninstall_shadows is true any differing versions throughout @INC will be uninstalled.

This is "make install UNINST=1"

As of 1.37_02 `install()` supports the use of a list of patterns to filter out files that shouldn't be installed. If `$skip` is omitted or undefined then `install` will try to read the list from `INSTALL.SKIP` in the CWD. This file is a list of regular expressions and is just like the `MANIFEST.SKIP` file used by `ExtUtils::Manifest`.

A default site `INSTALL.SKIP` may be provided by setting the environment variable `EU_INSTALL_SITE_SKIPFILE`, this will only be used when there isn't a distribution specific `INSTALL.SKIP`. If the environment variable `EU_INSTALL_IGNORE_SKIP` is true then no install file filtering will be performed.

If `$skip` is undefined then the skip file will be autodetected and used if it is found. If `$skip` is a reference to an array then it is assumed the array contains the list of patterns, if `$skip` is a true non reference it is assumed to be the filename holding the list of patterns, any other value of `$skip` is taken to mean that no install filtering should occur.

Changes As of Version 1.47

As of version 1.47 the following additions were made to the install interface. Note that the new argument style and use of the `%result` hash is recommended.

The `$always_copy` parameter which when true causes files to be updated regardless as to whether they have changed, if it is defined but false then copies are made only if the files have changed, if it is undefined then the value of the environment variable `EU_INSTALL_ALWAYS_COPY` is used as default.

The `%result` hash will be populated with the various keys/subhashes reflecting the install.

Currently these keys and their structure are:

```
install      => { $target => $source },
install_fail => { $target => $source },
install_unchanged => { $target => $source },
install_filtered => { $source => $pattern },
uninstall    => { $uninstalled => $source },
uninstall_fail => { $uninstalled => $source },
```

where `$source` is the filespec of the file being installed. `$target` is where it is being installed to, and `$uninstalled` is any shadow file that is in `@INC` or `$ENV{PERL5LIB}` or other standard locations, and `$pattern` is the pattern that caused a source file to be skipped. In future more keys will be added, such as to show created directories, however this requires changes in other modules and must therefore wait.

These keys will be populated before any exceptions are thrown should there be an error.

Note that all updates of the %result are additive, the hash will not be cleared before use, thus allowing status results of many installs to be easily aggregated.

NEW ARGUMENT STYLE

If there is only one argument and it is a reference to an array then the array is assumed to contain a list of key-value pairs specifying the options. In this case the option "from_to" is mandatory. This style means that you do not have to supply a cryptic list of arguments and can use a self documenting argument list that is easier to understand.

This is now the recommended interface to install().

RETURN

If all actions were successful install will return a hashref of the results as described above for the \$result parameter. If any action is a failure then install will die, therefore it is recommended to pass in the \$result parameter instead of using the return value. If the result parameter is provided then the returned hashref will be the passed in hashref.

install_default

DISCOURAGED

```
install_default();  
install_default($fullext);
```

Calls install() with arguments to copy a module from blib/ to the default site installation location.

\$fullext is the name of the module converted to a directory (ie. Foo::Bar would be Foo/Bar). If \$fullext is not specified, it will attempt to read it from @ARGV.

This is primarily useful for install scripts.

NOTE This function is not really useful because of the hard-coded install location with no way to control site vs core vs vendor directories and the strange way in which the module name is given. Consider its use discouraged.

uninstall

```
uninstall($packlist_file);  
uninstall($packlist_file, $verbose, $dont_execute);
```

Removes the files listed in a \$packlist_file.

If \$verbose is true, will print out each file removed. Default is false.

If \$dont_execute is true it will only print what it was going to do without actually doing

it. Default is false.

pm_to_blib

```
pm_to_blib(\%from_to);
```

```
pm_to_blib(\%from_to, $autosplit_dir);
```

```
pm_to_blib(\%from_to, $autosplit_dir, $filter_cmd);
```

Copies each key of %from_to to its corresponding value efficiently. If an \$autosplit_dir is provided, all .pm files will be autosplit into it. Any destination directories are created.

\$filter_cmd is an optional shell command to run each .pm file through prior to splitting and copying. Input is the contents of the module, output the new module contents.

You can have an environment variable PERL_INSTALL_ROOT set which will be prepended as a directory to each installed file (and directory).

By default verbose output is generated, setting the PERL_INSTALL_QUIET environment variable will silence this output.

ENVIRONMENT

PERL_INSTALL_ROOT

Will be prepended to each install path.

EU_INSTALL_IGNORE_SKIP

Will prevent the automatic use of INSTALL.SKIP as the install skip file.

EU_INSTALL_SITE_SKIPFILE

If there is no INSTALL.SKIP file in the make directory then this value can be used to provide a default.

EU_INSTALL_ALWAYS_COPY

If this environment variable is true then normal install processes will always overwrite older identical files during the install process.

Note that the alias EU_ALWAYS_COPY will be supported if EU_INSTALL_ALWAYS_COPY is not defined until at least the 1.50 release. Please ensure you use the correct

EU_INSTALL_ALWAYS_COPY.

AUTHOR

Original author lost in the mists of time. Probably the same as Makemaker.

Production release currently maintained by demerphq "yves at cpan.org", extensive changes by Michael G. Schwern.

Send bug reports via <http://rt.cpan.org/>. Please send your generated Makefile along with

your report.

LICENSE

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>>

perl v5.34.0

2023-11-23

ExtUtils::Install(3perl)