



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 'perlmoinstall.1'

\$ man perlmoinstall.1

PERLMODINSTALL(1) Perl Programmers Reference Guide PERLMODINSTALL(1)

NAME

perlmoinstall - Installing CPAN Modules

DESCRIPTION

You can think of a module as the fundamental unit of reusable Perl code; see perlmod for details. Whenever anyone creates a chunk of Perl code that they think will be useful to the world, they register as a Perl developer at <https://www.cpan.org/modules/04pause.html> so that they can then upload their code to the CPAN. The CPAN is the Comprehensive Perl Archive Network and can be accessed at <https://www.cpan.org/>, and searched at <https://metacpan.org/>. This documentation is for people who want to download CPAN modules and install them on their own computer.

PREAMBLE

First, are you sure that the module isn't already on your system? Try "perl -MFoo -e 1". (Replace "Foo" with the name of the module; for instance, "perl -MCGI::Carp -e 1".) If you don't see an error message, you have the module. (If you do see an error message, it's still possible you have the module, but that it's not in your path, which you can display with "perl -e "print qq(@INC)".) For the remainder of this document, we'll assume that you really honestly truly lack an installed module, but have found it on the CPAN. So now you have a file ending in .tar.gz (or, less often, .zip). You know there's a tasty module inside. There are four steps you must now take:

DECOMPRESS the file

UNPACK the file into a directory

BUILD the module (sometimes unnecessary)

INSTALL the module.

Here's how to perform each step for each operating system. This is <not> a substitute for reading the README and INSTALL files that might have come with your module!

Also note that these instructions are tailored for installing the module into your system's repository of Perl modules, but you can install modules into any directory you wish. For instance, where I say "perl Makefile.PL", you can substitute "perl Makefile.PL PREFIX=/my/perl_directory" to install the modules into /my/perl_directory. Then you can use the modules from your Perl programs with "use lib "/my/perl_directory/lib/site_perl";" or sometimes just "use "/my/perl_directory";". If you're on a system that requires superuser/root access to install modules into the directories you see when you type "perl -e "print qq(@INC)""", you'll want to install them into a local directory (such as your home directory) and use this approach.

? If you're on a Unix or Unix-like system,

You can use Andreas Koenig's CPAN module (<<https://metacpan.org/release/CPAN>>) to automate the following steps, from DECOMPRESS through INSTALL.

A. DECOMPRESS

Decompress the file with "gzip -d yourmodule.tar.gz"

You can get gzip from <<ftp://prep.ai.mit.edu/pub/gnu/>>

Or, you can combine this step with the next to save disk space:

```
gzip -dc yourmodule.tar.gz | tar -xof -
```

B. UNPACK

Unpack the result with "tar -xof yourmodule.tar"

C. BUILD

Go into the newly-created directory and type:

```
perl Makefile.PL
```

```
make test
```

or

```
perl Makefile.PL PREFIX=/my/perl_directory
```

to install it locally. (Remember that if you do this, you'll have to put "use lib "/my/perl_directory";" near the top of the program that is to use this module.

D. INSTALL

While still in that directory, type:

```
make install
```

Make sure you have the appropriate permissions to install the module in your Perl 5 library directory. Often, you'll need to be root.

That's all you need to do on Unix systems with dynamic linking. Most Unix systems have dynamic linking. If yours doesn't, or if for another reason you have a statically-linked perl, and the module requires compilation, you'll need to build a new Perl binary that includes the module. Again, you'll probably need to be root.

? If you're running ActivePerl (Win95/98/2K/NT/XP, Linux, Solaris),

First, type "ppm" from a shell and see whether ActiveState's PPM repository has your module. If so, you can install it with "ppm" and you won't have to bother with any of the other steps here. You might be able to use the CPAN instructions from the "Unix or Linux" section above as well; give it a try. Otherwise, you'll have to follow the steps below.

A. DECOMPRESS

You can use the open source 7-zip ([<https://www.7-zip.org/>](https://www.7-zip.org/)) or the shareware Winzip ([<https://www.winzip.com>](https://www.winzip.com)) to decompress and unpack modules.

B. UNPACK

If you used WinZip, this was already done for you.

C. BUILD

You'll need either "nmake" or "gmake".

Does the module require compilation (i.e. does it have files that end in .xs, .c, .h, .y, .cc, .cxx, or .C)? If it does, life is now officially tough for you, because you have to compile the module yourself (no easy feat on Windows). You'll need a compiler such as Visual C++. Alternatively, you can download a pre-built PPM package from ActiveState. [<http://aspn.activestate.com/ASPN/Downloads/ActivePerl/PPM/>](http://aspn.activestate.com/ASPN/Downloads/ActivePerl/PPM/)

Go into the newly-created directory and type:

```
perl Makefile.PL
```

```
nmake test
```

D. INSTALL

While still in that directory, type:

```
nmake install
```

? If you're on the DJGPP port of DOS,

A. DECOMPRESS

djtarx ([<ftp://ftp.delorie.com/pub/djgpp/current/v2/>](http://ftp.delorie.com/pub/djgpp/current/v2/)) will both uncompress and unpack.

B. UNPACK

See above.

C. BUILD

Go into the newly-created directory and type:

```
perl Makefile.PL  
make test
```

You will need the packages mentioned in README.dos in the Perl distribution.

D. INSTALL

While still in that directory, type:

```
make install
```

You will need the packages mentioned in README.dos in the Perl distribution.

? If you're on OS/2,

Get the EMX development suite and gzip/tar from Hobbes (<http://hobbes.nmsu.edu/h-browse.php?dir=/pub/os2/dev/emx/v0.9d>), and then follow the instructions for Unix.

? If you're on VMS,

When downloading from CPAN, save your file with a ".tgz" extension instead of ".tar.gz". All other periods in the filename should be replaced with underscores.

For example, "Your-Module-1.33.tar.gz" should be downloaded as "Your-Module-1_33.tgz".

A. DECOMPRESS

Type

```
gzip -d Your-Module.tgz
```

or, for zipped modules, type

```
unzip Your-Module.zip
```

Executables for gzip, zip, and VMStar:

```
http://www.hp.com/go/openvms/freeware/
```

and their source code:

```
http://www.fsf.org/order/ftp.html
```

Note that GNU's gzip/gunzip is not the same as Info-ZIP's zip/unzip package. The former is a simple compression tool; the latter permits creation of multi-file

archives.

B. UNPACK

If you're using VMStar:

```
VMStar xf Your-Module.tar
```

Or, if you're fond of VMS command syntax:

```
tar/extract/verbose Your_Module.tar
```

C. BUILD

Make sure you have MMS (from Digital) or the freeware MMK (available from MadGoat at <http://www.madgoat.com>). Then type this to create the DESCRIP.MMS for the module:

```
perl Makefile.PL
```

Now you're ready to build:

```
mms test
```

Substitute "mmk" for "mms" above if you're using MMK.

D. INSTALL

Type

```
mms install
```

Substitute "mmk" for "mms" above if you're using MMK.

? If you're on MVS,

Introduce the .tar.gz file into an HFS as binary; don't translate from ASCII to EBCDIC.

A. DECOMPRESS

Decompress the file with "gzip -d yourmodule.tar.gz"

You can get gzip from <http://www.s390.ibm.com/products/oe/bpxqp1.html>

B. UNPACK

Unpack the result with

```
pax -o to=IBM-1047,from=ISO8859-1 -r < yourmodule.tar
```

The BUILD and INSTALL steps are identical to those for Unix. Some modules generate

Makefiles that work better with GNU make, which is available from

<http://www.mks.com/s390/gnu/>

PORTABILITY

Note that not all modules will work with on all platforms. See perlport for more information on portability issues. Read the documentation to see if the module will work on your system. There are basically three categories of modules that will not work "out

of the box" with all platforms (with some possibility of overlap):

- ? Those that should, but don't. These need to be fixed; consider contacting the author and possibly writing a patch.
- ? Those that need to be compiled, where the target platform doesn't have compilers readily available. (These modules contain .xs or .c files, usually.) You might be able to find existing binaries on the CPAN or elsewhere, or you might want to try getting compilers and building it yourself, and then release the binary for other poor souls to use.
- ? Those that are targeted at a specific platform. (Such as the Win32:: modules.) If the module is targeted specifically at a platform other than yours, you're out of luck, most likely.

Check the CPAN Testers if a module should work with your platform but it doesn't behave as you'd expect, or you aren't sure whether or not a module will work under your platform.

If the module you want isn't listed there, you can test it yourself and let CPAN Testers know, you can join CPAN Testers, or you can request it be tested.

<https://cpantesters.org/>

HEY

If you have any suggested changes for this page, let me know. Please don't send me mail asking for help on how to install your modules. There are too many modules, and too few Orwants, for me to be able to answer or even acknowledge all your questions. Contact the module author instead, ask someone familiar with Perl on your operating system, or if all else fails, file a ticket at <<https://rt.cpan.org/>>.

AUTHOR

Jon Orwant

orwant@medita.mit.edu

with invaluable help from Chris Nandor, and valuable help from Brandon Allbery, Charles Bailey, Graham Barr, Dominic Dunlop, Jarkko Hietaniemi, Ben Holzman, Tom Horsley, Nick Ing-Simmons, Tuomas J. Lukka, Laszlo Molnar, Alan Olsen, Peter Prymmer, Gurusamy Sarathy, Christoph Spalinger, Dan Sugalski, Larry Virden, and Ilya Zakharevich.

First version July 22, 1998; last revised November 21, 2001.

COPYRIGHT

Copyright (C) 1998, 2002, 2003 Jon Orwant. All Rights Reserved.

This document may be distributed under the same terms as Perl itself.

