

NAME

mst-cpan – install CPAN modules in an alternative Perl root (plus more!)

SYNOPSIS

mst-cpan [options] package package

DESCRIPTION

mst-cpan installs Perl CPAN packages in an alternative Perl module directory hierarchy. More importantly it also encodes local preferences and fixes for various packages so that they will take advantage of local libraries.

OPTIONS AND ARGUMENTS

Options are specified using a getopt style interface. Long names are available when preceded with double hyphens `--`, in which case only the minimal number of characters are required. Options which take values may be separated from those values with the `=` character.

Arguments

Arguments (i.e. packages) are specified in one of three ways:

- As a CPAN Bundle file (ends in `.pm`)
 - `Local.pm`
- As a module name:
 - `Data::Dumper`
- As a local tarball:
 - `MyPackage-1.00.tar.gz`
- As a CPAN distribution
 - `AUTHOR/Package-1.00.tar.gz`

It's possible to confuse the latter two: is `AUTHOR` a directory name (indicating a local tarball) or a CPAN author id? To avoid mistakenly searching CPAN for packages because the user misspelled the tarball name **mst-cpan** uses the following heuristics to distinguish the two cases.

1. Packages which end in `.gz` are treated as local tarballs or CPAN distributions.
2. If a file of the given name exists, it is a local tarball. (Use the `--dist` option to override this behavior).
3. If the path contains leading “directories” all in captial letters, it is a CPAN distribution (Use the `--tarball` option to override this behavior.)
4. Fail.

Note that the `--dist` and `--tarball` options affect the behavior for all of the packages on the command line.

Directory Options**--install_base=directory**

The root of the directory hierarchy into which Perl modules are installed. Actual modules are typically installed in `install_base/lib/perl5`.

This defaults to `@ALTPERLROOT@/$platform_os_generic`.

--local_install_base=directory

If specified, the modules will be installed relative to the given base directory. Embedded `sysarch` or Perl `Config` variable names will be interpolated. See “Variable Interpolation” for more information.

`install_base` will still be used as a base modules directory to find modules. This allows one to have a private Perl module installation on top of the alternate maintained by **mst-cpan**.

The `--noprereqs` option may be useful if one wishes to install a locally produced module into `local_install_base` but not its CPAN prerequisites. There is no way to install CPAN only modules in `install_base` and other modules in `local_install_base` in the same invocation of **mst-cpan**.

Installation options**--dist**

Treat the package as a CPAN distribution, regardless of what the name looks like.

--force

Force the reinstallation of a module even if Perl thinks that it has already been installed. This is useful if the native Perl has had a module installed and you wish to reinstall it into the alternative module directory.

--force_install

Force the installation of a module even if tests have failed. Currently if tests succeeded this will result in a duplicate installation run.

--reindex

Cause **mst-cpan** to retrieve the latest CPAN index. It will usually do so automatically, but only after the current index is some few days old.

--patchdir

Some Perl modules require patches to get them to install cleanly. This directory contains patches entitled `module-version-patch`, where *module* is the module name, *version* is the module version id, and *patch* is the word `patch`. For some reason some modules' version ids are only recognized in a rewritten form, where the more common `X.Y.Z` is translated as `X.yyyzzz` where *yyy* and *zzz* are three digit versions of *Y* and *Z* with leading padding zeroes. For example, PDL version `2.4.3` is rewritten as `2.004003`

--prereqs, --noprereqs

Normally module prerequisites are installed according to the policy which the user requested when **cpan** was first run. Setting **--prereqs** forces a policy of `follow`, while **--noprereqs** forces the policy to `ignore`.

--tarball

Treat the package as a local tarball, regardless of what the name looks like.

Local library support

Some binary Perl modules may require libraries which are locally compiled and installed (i.e not in the standard system directories). Their location can be specified with the following options:

--prefix=directory

This is the top element of the path. It defaults to `/proj/axaf`.

--exec_prefix=directory

The architecture-dependent path. It defaults to `/prefix/platform_os_generic` where *prefix* is provided by the **--prefix** option and *platform_os_generic* is a directory specific to the platform that **mst-cpan** is being run on.

--libdir=directory

The directory containing the libraries. Embedded **sysarch** or Perl **Config** variable names will be interpolated. See Variable Interpolation for more information. It defaults to `/exec_prefix/lib` where *execprefix* is provided by the **--exec_prefix** option.

--bindir=directory

A directory holding locally installed executables, used to find executables associated with particular libraries. Embedded **sysarch** or Perl **Config** variable names will be interpolated. See Variable Interpolation for more information. It defaults to `/exec_prefix/bin` where *execprefix* is provided by the **--exec_prefix** option.

Miscellaneous**--cdebug**

Set the CPAN debug parameter. This takes an optional CPAN debug code. If not specified it will be set to `65535`.

- debug**
- help**
Output a short help message and exit.
- usage**
Output a longer usage message and exit.
- version**
Output the version and exit.

Variable Interpolation

The values for some of the parameters may contain references to **sysarch** or Perl **Config** variables. A variable reference is formed either by simply prefixing the variable name with a `$` or by enclosing the name in curly braces and prefixing that with a `$`:

```
$foo
${foo}
```

The names of variables provided by **sysarch** are used as is. Those provided by the Perl **Config** module must be prefixed with the string `perl_`,

```
$platform_os_generic
$perl_version
```

The available Perl **Config** variables may be listed via the following command:

```
perl -MConfig -MData::Dumper -e 'print Dumper \%Config'
```

Bootstrapping mst-cpan

1. The **CPAN** module must already be installed. Some Linux distributions package this separately from the core Perl package.
2. A version of **ExtUtils::MakeMaker** at least as new as 6.31 must be available. If the installed version is too old, the following installation procedure should be followed:
 1. Download the package from CPAN, decompress it, and change into the resulting directory.
 2. Run the following commands:

```
perl Makefile.PL
make
```

```
perl -Iblib Makefile.PL INSTALL_BASE=@ALTPERLROOT@/$platform_os_generic
make
make test
make install
```

Where `$platform_os_generic` is the **sysarch** variable of the same name.

3. Run **mst-cpan** with the **--setup** flag:

```
mst-cpan --setup [libdir]
```

where *libdir* is an optional argument specifying a path in which Perl should search for the **CPAN** module.

COPYRIGHT AND LICENSE

This software is Copyright The Smithsonian Astrophysical Observatory and is released under the GNU General Public License. You may find a copy at: <http://www.fsf.org/copyleft/gpl.html>

VERSION

This documents version @VERSION@ of @PACKAGE@.

AUTHOR

Diab Jerius <djerius@cfa.harvard.edu>