file: perlbrew.md

- metacpan install and run overview
- using-perlbrew-on-ubuntu
- perlbrew.pl
- good perlbrew overview
- Comprehensive Perl Archive Network
- www.perl.org
- How to install, remove or purge and uninstall perlbrew On Ubuntu 16.04
- How to install perl modules through CPAN on Ubuntu

Description

perlbrew is used to install manage perl installation in a users home directory. In the other words, perlbrew is an admin-free perl installation management tool.

installing perlbrew

install, initialize and list available perl versions

```
# Install requirements. We will use curl in order to install perlbrew.
sudo apt-get install  -y curl

# Installing perlbrew to our home directory
curl -L https://install.perlbrew.pl |  bash

# source
#
# setup your perlbrew environment (paths and some perlbrew commands)
source ~/perl5/perlbrew/etc/bashrc

# Initialize your installation, this only needs to be done once
perlbrew init

# test you environment, is perlbrew working correctly?
perlbrew

# See what versions of perl are available
perlbrew available
```

Output example

```
  perl-5.27.0
  perl-5.26.0
  perl-5.24.1
  perl-5.22.3
  perl-5.20.3
  perl-5.18.4
  perl-5.16.3
  perl-5.14.4
  perl-5.12.5
  perl-5.10.1
  perl-5.8.9
  perl-5.6.2
  perl5.005_04
  perl5.004_05
  cperl-5.27.0
```

Day to day usage

```
# Set up perlbrews environment if you have not added it to a startup file
source ~/perl5/perlbrew/etc/bashrc
```

install a version of perl

This is included because perl installations compiled from scratch using perlbrew fail often. See the section titled **Installing a perl using the `--notest` option**" below for a way to install this version of perl when the included tests fail..

```
perlbrew install 5.14.2
```

By opening a second terminal session and running the following command you can watch the installation progress by using the `tail` command:

```
tail -f ~/perl5/perlbrew/build.perl-5.14.2.log
```

The installation failed on my system (this happens frequently) and using the `cat` command I see the details:

```
cat /home/vagrant/perl5/perlbrew/build.perl-5.14.2.log
```

The end of our log:

```
Test Summary Report
-------------------
op/numconvert.t                                              (Wstat:   0 Tests: 1444
Failed: 12)
  Failed tests:  104, 108, 112, 136, 140, 144, 152, 156
                 160, 168, 172, 176
op/range.t                                                   (Wstat:   0 Tests: 141
Failed: 25)
  Failed tests:  84-95, 99-111
../lib/h2ph.t                                                (Wstat:   0 Tests: 6
Failed: 2)
  Failed tests:  5-6
Files=2085, Tests=454853, 446 wallclock secs (16.92 usr   5.64 sys + 161.26 cusr 13.72
csys = 197.54 CPU)
Result: FAIL
makefile:947: recipe for target 'test_harness' failed
make: *** [test_harness] Error 39
##### Brew Failed #####
```

Installing a perl using the --notest option

Here we attempt an installation of the same version using the `--notest` option:

```
perlbrew --notest install 5.14.2
```

In this case my perl works as the included tests where not run. I `cat` my log file again:

```
cat /home/vagrant/perl5/perlbrew/build.perl-5.14.2.log
```

and at the end I see that brew has finished the installation:

Output example:

```
...
  /home/vagrant/perl5/perlbrew/perls/perl-5.14.2/man/man1/find2perl.1
  /home/vagrant/perl5/perlbrew/perls/perl-5.14.2/man/man1/s2p.1
  /home/vagrant/perl5/perlbrew/perls/perl-5.14.2/man/man1/psed.1
make[1]: Leaving directory '/home/vagrant/perl5/perlbrew/build/perl-5.14.2'
##### Brew Finished #####
```

Installing with the --for ce flag

In cases where the `--notest` flag does not work you could also try to force the installation running the following command:

```
perlbrew --force install 5.14.2
```

Setting a ne w default perl version for your user

The default version of perl for all users on your system is the version of perl that gets installed when you run `sudo apt-get install perl`. This version of perl is the one that gets executed when you, or a program, runs a perl command. This version of perl gets run because it is the first executable file called `perl` on your **path**.

First lets take a look at where the systems default version of perl is located using the `which` command:

```
which perl
```

output:

```
/usr/bin/perl
```

Next lets look at your users path variable using the echo command followed by `$` and the PATH variable name.

```
echo $PATH
```

Output example:

```
/home/csteel/bin:/home/csteel/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr
/bin:/sbin:/bin:/usr/games:/usr/local/games
```

Next lets take a look at the location of the version of perl that is installed on the system when someone runs `sudo apt-get install -y perl`.

```
which perl
```

output:

```
/usr/bin/perl
```

keeping in mind that each path defined in your users PATH variable is separated using colon `:` you can see that directory containing the systems version of perl, in this case `/usr/bin`, is located nearer to the end of my path just before `/sbin`.

setting a new default version of perl using perlbrew

First I will list the versions of perl available to me in my home using perlbrew.

```
perlbrew list
```

example output:

```
   perl-5.14.2
```

I only have one, **5.14.2**. I am going to make it my default version of perl by running the following command:

```
perlbrew switch perl-5.14.2
```

testing

Next I will run some tests to ensure that my default version of perl has changed. First I am going to run the `which` command again:

```
which perl
```

Notice that the first perl on my path, the one that gets displayed when I type `which perl` is now in my home directory here:

```
/home/csteel/perl5/perlbrew/perls/perl-5.14.2/bin/perl
```

by taking a look at the value of my PATH variable using

```
echo $PATH
```

I see that the directory `/home/csteel/perl5/perlbrew/perls/perl-5.14.2/bin` , the directory that contains perl version 5.14.2 the second path in my PATH which means that the version of perl on this second path is the version of perl that gets executed rather than the system version located at `/usr/bin/perl. is now at the beginning of my path.

```
home/csteel/perl5/perlbrew/bin:/home/csteel/perl5/perlbrew/perls/perl-
5.14.2/bin:/home/csteel/bin:/home/csteel/.local/bin:/usr/local/sbin:/usr/local/bin:/us
r/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```