

NAME

CPANPLUS::Config

SYNOPSIS

```
### conf object via CPANPLUS::Backend;
$cb  = CPANPLUS::Backend->new;
$conf = $cb->configure_object;

### or as a standalone object
$conf = CPANPLUS::Configure->new;

### values in 'conf' section
$verbose = $conf->get_conf( 'verbose' );
$conf->set_conf( verbose => 1 );

### values in 'program' section
$editor = $conf->get_program( 'editor' );
$conf->set_program( editor => '/bin/vi' );
```

DESCRIPTION

This module contains defaults and heuristics for configuration information for CPANPLUS. To change any of these values, please see the documentation in `CPANPLUS::Configure`.

Below you'll find a list of configuration types and keys, and their meaning.

CONFIGURATION

Section 'conf'

hosts

An array ref containing hosts entries to be queried for packages.

An example entry would like this:

```
{  'scheme' => 'ftp',
   'path'  => '/pub/CPAN/',
   'host'  => 'ftp.cpan.org'
},
```

allow_build_interactivity

Boolean flag to indicate whether 'perl Makefile.PL' and similar are run interactively or not. Defaults to 'true'.

base

The directory CPANPLUS keeps all its build and state information in. Defaults to `~/.cpanplus`.

buildflags

Any flags to be passed to 'perl Build.PL'. See `perldoc Module::Build` for details. Defaults to an empty string.

cpantest

Boolean flag to indicate whether or not to mail test results of module installations to `http://testers.cpan.org`. Defaults to 'false'.

cpantest_mx

String holding an explicit mailserver to use when sending out emails for

`http://testers.cpan.org`. An empty string will use your system settings. Defaults to an empty string.

debug

Boolean flag to enable or disable extensive debuggging information. Defaults to 'false'.

dist_type

Default distribution type to use when building packages. See `cpan2dist` or `CPANPLUS::Dist` for details. An empty string will not use any package building software. Defaults to an empty string.

email

Email address to use for anonymous ftp access and as `from` address when sending emails. Defaults to an `example.com` address.

enable_custom_sources

Boolean flag indicating whether custom sources should be enabled or not. See the `CUSTOM MODULE SOURCES` in `CPANPLUS::Backend` for details on how to use them. Defaults to `true`

extractdir

String containing the directory where fetched archives should be extracted. An empty string will use a directory under your `base` directory. Defaults to an empty string.

fetchdir

String containing the directory where fetched archives should be stored. An empty string will use a directory under your `base` directory. Defaults to an empty string.

flush

Boolean indicating whether build failures, cache dirs etc should be flushed after every operation or not. Defaults to 'true'.

force

Boolean indicating whether files should be forcefully overwritten if they exist, modules should be installed when they fail tests, etc. Defaults to 'false'.

lib

An array ref holding directories to be added to `@INC` when CPANPLUS starts up. Defaults to an empty array reference.

makeflags

A string holding flags that will be passed to the `make` program when invoked. Defaults to an empty string.

makemakerflags

A string holding flags that will be passed to `perl Makefile.PL` when invoked. Defaults to an empty string.

md5

A boolean indicating whether or not md5 checks should be done when an archive is fetched. Defaults to 'true' if you have `Digest::MD5` installed, 'false' otherwise.

no_update

A boolean indicating whether or not CPANPLUS' source files should be updated or not. Defaults to 'false'.

passive

A boolean indicating whether or not to use passive ftp connections. Defaults to 'true'.

prefer_bin

A boolean indicating whether or not to prefer command line programs over perl modules. Defaults to 'false' unless you do not have `Compress::Zlib` installed (as that would mean we could not extract `.tar.gz` files)

prefer_makefile

A boolean indicating whether or not prefer a `Makefile.PL` over a `Build.PL` file if both are present. Defaults to 'true'.

prereqs

A digit indicating what to do when a package you are installing has a prerequisite. Options are:

```
0    Do not install
1    Install
2    Ask
3    Ignore   (dangerous, install will probably fail!)
```

The default is to ask.

shell

A string holding the shell class you wish to start up when starting CPANPLUS in interactive mode.

Defaults to `CPANPLUS::Shell::Default`, the default CPANPLUS shell.

show_startup_tip

A boolean indicating whether or not to show start up tips in the interactive shell. Defaults to 'true'.

signature

A boolean indicating whether or not check signatures if packages are signed. Defaults to 'true' if you have `gpg` or `Crypt::OpenPGP` installed, 'false' otherwise.

skiptest

A boolean indicating whether or not to skip tests when installing modules. Defaults to 'false'.

storable

A boolean indicating whether or not to use `Storable` to write compiled source file information to disk. This makes for faster startup and look up times, but takes extra disk space. Defaults to 'true' if you have `Storable` installed and 'false' if you don't.

timeout

Digit indicating the time before a fetch request times out (in seconds). Defaults to 300.

verbose

A boolean indicating whether or not CPANPLUS runs in verbose mode. Defaults to 'true' if you have the environment variable `PERL5_CPANPLUS_VERBOSE` set to true, 'false' otherwise.

It is recommended you run with verbose enabled, but it is disabled for historical reasons.

write_install_log

A boolean indicating whether or not to write install logs after installing a module using the interactive shell. Defaults to 'true'.

source_engine

Class to use as the source engine, which is generally a subclass of `CPANPLUS::Internals::Source`. Default to `CPANPLUS::Internals::Source::Memory`.

`cpantest_reporter_args`

A hashref of key => value pairs that are passed to the constructor of `Test::Reporter`. If you'd want to enable TLS for example, you'd set it to:

```
{ transport      => 'Net::SMTP::TLS',  
  transport_args => [ User => 'Joe', Password => '123' ],  
}
```

Section 'program'

`editor`

A string holding the path to your editor of choice. Defaults to your `$ENV{EDITOR}`, `$ENV{VISUAL}`, `'vi'` or `'pico'` programs, in that order.

`make`

A string holding the path to your `make` binary. Looks for the `make` program used to build perl or failing that, a `make` in your path.

`pager`

A string holding the path to your pager of choice. Defaults to your `$ENV{PAGER}`, `'less'` or `'more'` programs, in that order.

`shell`

A string holding the path to your login shell of choice. Defaults to your `$ENV{SHELL}` setting, or `$ENV{COMSPEC}` on Windows.

`sudo`

A string holding the path to your `sudo` binary if your install path requires super user permissions. Looks for `sudo` in your path, or remains empty if you do not require super user permissions to install.

`perlwrapper`

A string holding the path to the `cpanp-run-perl` utility bundled with CPANPLUS, which is used to enable autoflushing in spawned processes.

BUG REPORTS

Please report bugs or other issues to <bug-cpanplus@rt.cpan.org>.

AUTHOR

This module by Jos Boumans <kane@cpan.org>.

COPYRIGHT

The CPAN++ interface (of which this module is a part of) is copyright (c) 2001 - 2007, Jos Boumans <kane@cpan.org>. All rights reserved.

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

SEE ALSO

CPANPLUS::Backend, *CPANPLUS::Configure::Setup*, *CPANPLUS::Configure*