NAME

CPANPLUS::Configure

SYNOPSIS

```
    $conf   = CPANPLUS::Configure->new( );

    $bool   = $conf->can_save;
    $bool   = $conf->save( $where );

    @opts   = $conf->options( $type );

    $make       = $conf->get_program('make');
    $verbose    = $conf->set_conf( verbose => 1 );
```

DESCRIPTION

This module deals with all the configuration issues for CPANPLUS. Users can use objects created by this module to alter the behaviour of CPANPLUS.

Please refer to the `CPANPLUS::Backend` documentation on how to obtain a `CPANPLUS::Configure` object.

METHODS

$Configure = CPANPLUS::Configure->new(load_configs => BOOL)

This method returns a new object. Normal users will never need to invoke the `new` method, but instead retrieve the desired object via a method call on a `CPANPLUS::Backend` object.

The `load_configs` parameter controls wether or not additional user configurations are to be loaded or not. Defaults to `true`.

$bool = $Configure->init([rescan => BOOL])

Initialize the configure with other config files than just the default 'CPANPLUS::Config'.

Called from `new()` to load user/system configurations

If the `rescan` option is provided, your disk will be examined again to see if there are new config files that could be read. Defaults to `false`.

Returns true on success, false on failure.

can_save([$config_location])

Check if we can save the configuration to the specified file. If no file is provided, defaults to your personal config.

Returns true if the file can be saved, false otherwise.

$file = $conf->save([$package_name])

Saves the configuration to the package name you provided. If this package is not `CPANPLUS::Config::System`, it will be saved in your `.cpanplus` directory, otherwise it will be attempted to be saved in the system wide directory.

If no argument is provided, it will default to your personal config.

Returns the full path to the file if the config was saved, false otherwise.

options(type => TYPE)

Returns a list of all valid config options given a specific type (like for example `conf` of `program`) or false if the type does not exist

ACCESSORS

Accessors that start with a _ are marked private -- regular users should never need to use these.

See the `CPANPLUS::Config` documentation for what items can be set and retrieved.

get_SOMETHING(ITEM, [ITEM, ITEM, ...]);

The `get_*` style accessors merely retrieves one or more desired config options.

set_SOMETHING(ITEM => VAL, [ITEM => VAL, ITEM => VAL, ...]);

The `set_*` style accessors set the current value for one or more config options and will return true upon success, false on failure.

add_SOMETHING(ITEM => VAL, [ITEM => VAL, ITEM => VAL, ...]);

The `add_*` style accessor adds a new key to a config key.

Currently, the following accessors exist:

set|get_conf

Simple configuration directives like verbosity and favourite shell.

set|get_program

Location of helper programs.

_set|_get_build

Locations of where to put what files for CPANPLUS.

_set|_get_source

Locations and names of source files locally.

_set|_get_mirror

Locations and names of source files remotely.

_set|_get_fetch

Special settings pertaining to the fetching of files.

BUG REPORTS

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

AUTHOR

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

COPYRIGHT

SEE ALSO

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