NAME

ProjectBuilder::Base, part of the project−builder.org − module dealing with generic functions suitable for perl project development

DESCRIPTION

This module provides generic functions suitable for perl project development

SYNOPSIS

```
  use ProjectBuilder::Base;

  #
  # Create a directory and its parents
  #
  pb_mkdir_p("/tmp/foo/bar");

  #
  # Remove recursively a directory and its children
  #
  pb_rm_rf("/tmp/foo");

  #
  # Encapsulate the system call for better output and return value test
  #
  pb_system("ls -l", "Printing directory content");

  #
  # Analysis a URI and return its components in a table
  #
  my ($scheme, $account, $host, $port, $path) = pb_get_uri("svn+ssh://ac@my.server.o

  #
  # Gives the current date in a table
  #
  @date = pb_get_date();

  #
  # Manages logs of the program
  #
  pb_log_init(2,\*STDOUT);
  pb_log(1,"Message to print\n");

  #
  # Manages content of a file
  #
  pb_display_file("/etc/passwd",\*STDERR);
  my $cnt = pb_get_content("/etc/passwd");
```

USAGE

pb_mkdir_p

Internal mkdir −p function. Forces mode to 755. Supports multiple parameters.

Based on File::Path mkpath.

pb_rm_rf

Internal rm −rf function. Supports multiple parameters.

Based on File::Path rmtree.

pb_system

Encapsulate the "system" call for better output and return value test. Needs a `$ENV{'PBTMP'}` variable which is created by calling the pb_mktemp_init function. Needs pb_log support, so pb_log_init should have been called before.

The first parameter is the shell command to call. This command should NOT use redirections. The second parameter is the message to print on screen. If none is given, then the command is printed. The third parameter prints the result of the command after correct execution if value is "verbose". If value is "noredir", it avoids redirecting outputs (e.g. for vi). If value is "quiet", doesn't print anything at all. If value is "mayfail", failure of the command is ok even if `$Global::pb_stop_on_error` is set, because the caller will be handling the error. A "verbose" can be added to mayfail to have it explain why it failed. If value is verbose_PREF, then every output command will be prefixed with PREF. This function returns as a result the return value of the system command.

If no error reported, it prints OK on the screen, just after the message. Else it prints the errors generated.

pb_get_uri

This function returns a list of 6 parameters indicating the protocol, account, password, server, port, and path contained in the URI passed in parameter.

A URI has the format protocol://[ac@]host[:port][path[?query][#fragment]].

Cf man URI.

pb_get_date

This function returns a list of 9 parameters indicating the seconds, minutes, hours, day, month, year, day in the week, day in the year, and daylight saving time flag of the current time.

Cf: man ctime and description of the struct tm.

pb_log_init

This function initializes the global variables used by the pb_log function.

The first parameter is the debug level which will be considered during the run of the program? The second parameter is a pointer on a file descriptor used to print the log info.

As an example, if you set the debug level to 2 in that function, every call to pb_log which contains a value less or equal than 2 will be printed. Calls with a value greater than 2 won't be printed.

The call to **pb_log_init** is typically done after getting a parameter on the CLI indicating the level of verbosity expected.

pb_log

This function logs the messages passed as the second parameter if the value passed as first parameter is lesser or equal than the value passed to the **pb_log_init** function.

Here is a usage example:

```
pb_log_init(2,\*STDERR);
pb_log(1,"Hello World 1\n");
pb_log(2,"Hello World 2\n");
pb_log(3,"Hello World 3\n");

will print:

Hello World 1
Hello World 2
```

pb_display_file

This function prints the content of the file passed in parameter. If a second parameter is given, this is the descriptor of the logfile to write to in addtion to STDOUT. If a third parameter is given, this is the prefix providing it's writen as verbose_PREF. In which case the PREF string will be added before the line output.

This is a cat equivalent function.

pb_get_content
This function returns the content of the file passed in parameter.

pb_set_content
This function put the content of a variable passed as second parameter into the file passed as first parameter.

pb_exit
Fundtion to call before exiting pb so cleanup is done

pb_syntax_init
This function initializes the global variable used by the pb_syntax function.

The parameter is the message string which will be printed when calling pb_syntax

pb_syntax
This function prints the syntax expected by the application, based on pod2usage, and exits. The first parameter is the return value of the exit. The second parameter is the verbosity as expected by pod2usage.

Cf: man Pod::Usage

pb_temp_init
This function initializes the environemnt variable PBTMP to a random value. This directory can be safely used during the whole program, it will be removed at the end automatically.

pb_get_osrelease
This function returns the release of our operating system

pb_get_arch
This function returns the architecture of our local environment and standardize on i386 for those platforms.

pb_check_requirements
This function checks that the commands needed for the subsystem are indeed present. The required commands are passed as a comma separated string as first parameter. The optional commands are passed as a comma separated string as second parameter.

pb_check_req
This function checks existence of a command and return its full pathname or undef if not found. The command name is passed as first parameter. The second parameter should be 0 if the command is mandatory, 1 if optional. It returns the full path name of the command if found, undef otherwise and dies if that was a mandatory command

pb_path_expand
Expand out a path by environment variables as ($ENV{'ENVVAR'}) and ~

pb_path_nbfiles
Return the number of files in a directory

WEB SITES

The main Web site of the project is available at <http://www.project−builder.org/>. Bug reports should be filled using the trac instance of the project at <http://trac.project−builder.org/>.

USER MAILING LIST

None exists for the moment.

AUTHORS

The Project−Builder.org team <http://trac.project−builder.org/> lead by Bruno Cornec <mailto:bruno@project−builder.org>.

COPYRIGHT