## NAME

ProjectBuilder::Conf, part of the project−builder.org − module dealing with configuration files

## DESCRIPTION

This modules provides functions dealing with configuration files.

## SYNOPSIS

```
use ProjectBuilder::Conf;

#
# Read hash codes of values from a configuration file and return table of pointers
#
my ($k1, $k2) = pb_conf_read_if("$ENV{'HOME'}/.pbrc","key1","key2");
my ($k) = pb_conf_read("$ENV{'HOME'}/.pbrc","key");
```

## USAGE

The configuration files are loaded in a specific order from most generic to the most specific to allow for overwrite to work:

1. /usr/share/pb/pb.conf − the read-only system conf file provided by install 2. /etc/pb/pb.conf − the same global conf file given to the sysadmin in order to make system wide modifications 3. /path/to/project.pb − Configuration file for the project we're building for 4. /(vm|ve|rm)path/to/.pbrc − configuration file for VM, VE or RM specific parameters. Cumulative should be orthogonal 5. $HOME/.pbrc − user's configuration file

**pb_conf_init**

This function setup the environment pb for project-builder function usage from other projects. The first parameter is the project name. It sets up environment variables (pb)

**pb_conf_cache**

This function caches the configuration file content passed as first parameter into the a hash passed in second parameter It returns the modified hash Can be used in correlation with the %h hash to store permanently values or not if temporarily.

**pb_conf_add**

This function adds the configuration file to the list last, and cache their content in the %h hash

**pb_conf_read_if**

This function returns a table of pointers on hashes corresponding to the keys in a configuration file passed in parameter. If that file doesn't exist, it returns undef.

The format of the configuration file is as follows:

key tag = value1,value2,...

Supposing the file is called "$ENV{'HOME'}/.pbrc", containing the following:

```
$ cat $HOME/.pbrc
pbver pb = 3
pbver default = 1
pblist pb = 12,25
```

calling it like this:

```
my ($k1, $k2) = pb_conf_read_if("$ENV{'HOME'}/.pbrc","pbver","pblist");
```

will allow to get the mapping:

```
$k1−>{'pb'}  contains 3
$k1−>{'default'} contains 1
$k2−>{'pb'} contains 12,25
```

Valid chars for keys and tags are letters, numbers, '−' and '_'.

The file read is forgotten after its usage. If you want permanent caching of the data, use pb_conf_add then pb_conf_get

**pb_conf_read**

This function is similar to **pb_conf_read_if** except that it dies when the file in parameter doesn't exist.

**pb_conf_write**

This function writes in the file passed ias first parameter the hash of values passed as second parameter

**pb_conf_get_in_hash_if**

This function returns a table, corresponding to a set of values queried in the hash passed in parameter or undef if it doesn't exist. It takes a table of keys as an input parameter.

**pb_conf_get_if**

This function returns a table, corresponding to a set of values queried in the %h hash or undef if it doen't exist. It takes a table of keys as an input parameter.

The format of the configurations file is as follows:

key tag = value1,value2,...

It will gather the values from all the configurations files passed to pb_conf_add, and return the values for the keys

```
$ cat $HOME/.pbrc
pbver pb = 1
pblist pb = 4
$ cat $HOME/.pbrc2
pbver pb = 3
pblist default = 5
```

calling it like this:

```
pb_conf_add("$HOME/.pbrc","$HOME/.pbrc2");
my ($k1, $k2) = pb_conf_get_if("pbver","pblist");
```

will allow to get the mapping:

```
$k1−>{'pb'} contains 3
$k2−>{'pb'} contains 4
```

Valid chars for keys and tags are letters, numbers, '−' and '_'.

**pb_conf_add_last_in_hash**

This function merges the values passed in the hash parameter into the %h hash, but only if itdoesn't already contain a value, or if the value is more precise (real value instead of default)

It is used internally by pb_conf_add and is not exported.

**pb_conf_get**

This function is the same **pb_conf_get_if**, except that it tests each returned value as they need to exist in that case.

**pb_conf_get_all**

This function returns an array with all configuration parameters

**pb_conf_get_hash**

This function returns a pointer to the hash with all configuration parameters

## 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

Project−Builder.org is distributed under the GPL v2.0 license described in the file `COPYING` included with the distribution.