NAME
Documentation for pb configuration files

DESCRIPTION
pb helps you build various packages directly from your project sources. In order to work correctly, it relies on a certain number of configuration files. Most of these configuration parameters can be setup in all the configuration files, however, they generally make more sense in a specific one, which is indicated. There are mainly 4 configuration files, the one in the home directory of the user (used first), the one from the project (use in second), the one in the VM/VE hosting directory, and the one provided by the tool in /etc/pb or /usr/local/etc/pb (lastly).

SYNOPSIS
Those files have the same format, which is YAML starting after v0.14 of pb.
keyword:
  key: value1[,value2,...]
(Before it was using: keyword key = value1[,value2,...])
The key could be also default, in which case it will be used as a default value if no more precise content is given for the key.

Each value is detailed below giving the nature of its use (Mandatory or Optional – only used for certain feature), the value of the key (could be the project, an OS name, default, ...), the value of the parameter field and its format, the default configuration file in which it should be defined (home $HOME/.pbrc.yml, conf /etc/pb/pb.yml or /usr/local/etc/pb/pb.yml, VE vepath/.pbrc.yml, VM vmpath/.pbrc.yml, or project project.yml) and an example of use.

OPTIONS
addbuildrepo
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: comma separated list of URLs that point to repository files, or packages to install at project build time. The values may not include substitutions.
Conffile: project
Example: addbuildrepo:
  centos-5-x86_64: file:///prj/extras.repo,http://mirror.centos.org/centos/5.8/extras/x86_64/RPMS/chrpath-0.13-3.el5.centos.x86_64.rpm

addinstallrepo
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: comma separated list of URLs that point to repository files, or packages to install at project installation time. The values may not include substitutions.
Conffile: project
Example: addinstallrepo:
  centos-7-x86_64: ftp://ftp.project-builder.org/centos/7/x86_64/pb.repo

addtestrepo
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: comma separated list of URLs that point to repository files, or packages to install at project test time. The values may not include substitutions.
Conffile: project
Example: addtestrepo:
  centos-7-x86_64: ftp://ftp.project-builder.org/centos/7/x86_64/pb.repo

cachedir
Nature: Optional
Key: pb project: rpmbootstrap|pbmkbm|pb
Value: Directory to cache temporary content for the relevant pb project.
Conffile: pb
Example: cachedir:
  rpbootstrap: /var/cache/rpmbootstrap
checkexclude
Nature: Optional
Key: package (as provided in defpkgdir or extpkgdir)
Value: comma separated list of OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch) that are excluded from the checkssh command (no build made for them). The OS name is generally used here.
Conffile: project
Example: checkexclude:
  pkg1: centos, lsb, solaris

cpandir
Nature: Optional
Key: project (as defined in the −p option or pb environment variable)
Value: CPAN Pause directory to upload new modules
Conffile: pb
Example: cpandir:
  default: incoming

cpanpasswd
Nature: Optional
Key: project (as defined in the −p option or pb environment variable)
Value: CPAN Pause user's password
Conffile: home
Example: cpanpasswd:
  default: mycomplicatedpwd

cpanpause
Nature: Optional
Key: project (as defined in the −p option or pb environment variable)
Value: CPAN Pause site to upload new modules
Conffile: pb
Example: cpanpause:
  default: pause.perl.org

cpansubdir
Nature: Optional
Key: project (as defined in the −p option or pb environment variable)
Value: CPAN Pause subdirectory in the user's area to upload into
Conffile: pb
Example: cpansubdir:
  default: mydir

cpanurl
Nature: Optional
Key: project (as defined in the −p option or pb environment variable)
Value: CPAN Pause URL to activate the upload mechanism
Conffile: pb
Example: cpanurl:
  default: http://pause.perl.org/pause/authenquery

cpanuser
Nature: Optional
Key: project (as defined in the −p option or pb environment variable)
Value: CPAN Pause user
Conffile: home
Example: cpanuser:
  default: XXX

defpkgdir
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: comma separated list of packages built by default in this project. When not using any package name as a parameter to pb commands, this list will be used.
Conffile: project
Example: defpkgdir:
  mondorescue: mondo,mindi NB: a default value is not really meaningful.

delivery
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: directory where to deliver packages once built for ftp/web access.
Conffile: project
Example: delivery:
  mondorescue: prod

dockeropt
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: List of the options to call docker with
Conffile: project
Example: dockeropt:
  default: --bip=172.16.42.1/16

dockerregistry
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: name of the docker registry to interact with if any
Conffile: project
Example: dockerregistry:
  mondorescue: localhost:5900/mondorescue

dockerrepository
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: name of the docker repository to interact with if any. It is mandatory if no dockerregistry is defined.
Conffile: project
Example: dockerrepository:
  mondorescue: localhost:5000/mondorescue

extpkgdir
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: comma separated list of packages built in addition in this project. When using the all package name as a parameter to pb commands, this list will be used, in addition to the defpkgdir list.
Conffile: project
Example: extpkgdir:
  mondorescue: mondo-doc,mindi-mindibusybox

filteredfiles
Nature: Optional
Key: package (as provided in defpkgdir or extpkgdir)
Value: comma separated list of files that will be filtered using the macro system from pb, during the creation of the compressed source tar files for this package. Their path is relative to the directory containing the package.
Conffile: project
Example: filteredfiles:
  mindi: rootfs/sbin/init,mindi,install.sh,doc/mindi.8

ftp_proxy
http_proxy
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: string indicating the proxy to use
Conffile: pb
Example: http_proxy:
  default: http://example.com:3128/

https_proxy
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: string indicating the proxy to use
Conffile: pb
Example: https_proxy:
  default: http://example.com:3128/

logcmd
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: internal (the application then handles the logging of what it finds useful) or the name of an application to launch to log context (e.g. sos, cfg2html, ...).
Conffile: pb
Example: logcmd:
  mageia: sos

logcmds
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: In case the B<logcmd> command is internal, a comma separated list of the commands whose trace execution is to be captured in order to log context.
Conffile: pb
Example: logcmds:
  mageia: mount,lsmod,esxcfg-module -l,df -T

logfiles
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: In case the B<logcmd> command is internal, a comma separated list of the files to capture in order to log context.
Conffile: pb
Example: logfiles:
  mageia: /etc/raidtab,/proc/cmdline,/proc/swaps

logopt
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: In case the B<logcmd> command is not internal, the options of the B<logcmd> application to launch to log context
Conffile: pb
Example: logopt:
  mageia: --all

mkbmbotcmds
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). The family name is generally used here.
Value: comma separated list of commands to be copied from the original OS to the target boot media tree (works recursively for directory creation on the target).
Conffile: pb
Example: mkbmbootcmds:
   linux: perl, awk, gawk, dd, grep, uname

mkbmbootdirs
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). The family name is generally used here.
Value: comma separated list of directories to be copied from the original OS to the target boot media tree (works recursively on the target).
Conffile: pb
Example: mkbmbootdirs:
   linux: /etc/ssh, /etc/udev, /etc/mdadm

mkbmbootfiles
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). The family name is generally used here.
Value: comma separated list of files to be copied from the original OS to the target boot media tree (works recursively for directory creation on the target).
Conffile: pb
Example: mkbmbootfiles:
   linux: /etc/mdadm.conf, /etc/raidtab, /etc/modprobe.conf

mkbmkerneldir
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). The family name is generally used here.
Value: path of the directory containing your kernel.
Conffile: pb
Example: mkbmkerneldir:
   linux: /boot

mkbmkernelfile
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). The family name is generally used here.
Value: full path of the your kernel.
Conffile: pb
Example: mkbmkernelfile:
   linux: /boot/vmlinuz−specific

mkbmkernelnamere
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). The family name is generally used here.
Value: Perl Regular Expression allowing to find OS kernel names in the B<kerneldir> directory.
Conffile: pb
Example: mkbmkernelnamere:
   linux: ^linu|^vmlinu|^xen

mkbmtargetdirs
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). The family name is generally used here.
Value: comma separated list of empty directory paths to be created on the target.
Conffile: pb
Example: mkbmtargetdirs:
   linux: /tmp, /dev

namingtype
Nature: Optional
Key: package (as provided in defpkgdir or extpkgdir)
Value: perl, if packages are CPAN perl modules that need to be named respecting the distribution perl convention (perl−Name for rpm, libname−perl for deb)
Conffile: project
Example: namingtype:
  ProjectBuilder: perl

osambiguous
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). In that case, should be an OS name.
Value: comma separated list of distributions which use the same file name in /etc for different distributions (ex: /etc/redhat−release for redhat, rhel, mandriva,...). Cf: B<osrelambfile>
Conffile: pb
Example: osambiguous:
  debian: debian,ubuntu

oschkcmd
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). In that case, should be an OS name.
Value: package checker tool.
Conffile: pb
Example: oschkcmd:
  deb: /usr/bin/lintian

oschkopt
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). In that case, should be an OS name.
Value: package checker tool options.
Conffile: pb
Example: oschkcmd:
  rpm: -i

oscmd
Nature: Mandatory
Key: tool (pb or rpmbootstrap)
Value: coma separated list of commands that are mandatory on the underlying system
Conffile: pb
Example: oscmd:
  pb: tar,ls

oscmdopt
Nature: Mandatory
Key: tool (pb or rpmbootstrap)
Value: come separated list of commands that are optional on the underlying system
Conffile: pb
Example: oscmd:
  pb: svn2cl,svn,cvs

oscodename
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). In that case, should be an os−ver.
Value: code name. Mostly useful for debian and ubuntu, due to debootstrap.
Conffile: pb
Example: oscodename debian−5.0 = lenny

osfamily
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). Value: OS family name (used to group similar distribution for handling).
Conffile: pb
Example: osfamily:
  debian: du

osins
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). Value: OS command to launch in order to automatically install packages on it.
Conffile: pb
Example: osins:
  fedora: sudo yum -y install

oslocalins
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). Value: OS command to launch in order to automatically install local packages on it.
Conffile: pb
Example: oslocalins:
  debian: sudo dpkg -i

osmindep
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). Value: comma separated list of packages needed before setupvm|ve can be run. Install them with your distribution in VM or automatically in VE when possible.
Conffile: pb
Example: osmindep:
  default: perl,sudo,wget,tar,make,gzip

osnover
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). Value: true|false. True means that this OS has no need to keep the version.
Conffile: pb
Example: osnover:
  gentoo: true

ospatchcmd
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). Value: package patch command. For RPM is implicit.
Conffile: pb
Example: ospatchcmd:
  deb: /usr/bin/patch

ospatchopt
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). Value: package patch options.
Conffile: pb
Example: ospatchcmd:
  deb: -s -p1

ospatchcmd−*
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch)
Value: Full path name of the command mentioned after the '-' for the relative OS
Conffile: pb
Example: ospathcmd-halt:
   solaris: /usr/sbin/halt, ospathcmd-halt default = /sbin/halt

osperldep
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch)
Value: comma separated list of perl modules needed by pb and not provided in a package format for the relative OS, and that will be installed from CPAN in your VM/VE.
Conffile: pb
Example: osperldep:
   rhel-5: Module-Build,File-MimeInfo,File-BaseDir,Mail-Sendmail

osperlver
Nature: Mandatory (for each module mentioned in osperldep)
Key: Perl Module (as defined in osperldep)
Value: Version of the perl module that needs to be found on CPAN.
Conffile: pb
Example: osperlver:
   Date-Manip: 5.54

ospkg
Nature: Optional (Mandatory if pbinstalltype is pkg)
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch)
Value: comma separated list of packages that have to be installed in order for pb to be operational in the VE|VM
Conffile: pb
Example: ospkg:
   rhel-5: project-builder

ospkgdep
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch)
Value: comma separated list of packages that are needed by pb and should be installed automatically in the VM/VE during the setupvm|ve phase by pb.
Conffile: pb
Example: ospkgdep:
   rhel-5: wget,make,ntp,patch,perl-DateManip

osrelambfile
Nature: Mandatory (per OS mentioned in osambiguous)
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch)
Value: full path name of the ambiguous file describing that distribution, as well as some others. All the distributions mentioned here should also be mentioned with their ambiguous other distribution in the osambiguous parameter.
Conffile: pb
Example: osrelambfile:
   debian: /etc/debian_version

osrelexpr
Nature: Mandatory (per OS mentioned in osrelambfile and osrelfile)
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch)
Value: the perl regular expression used to parse the osrelambfile in order to extract from it the version of the distribution (in parenthesis to allow its usage as $1 by perl)
Conffile: pb
Example: osrelexpr:
   rhel: Red Hat (?:Enterprise Linux|Linux Advanced Server) .*release ([0-9.]+) (*
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-Ver, os-Ver-arch).
Value: full path name of the file describing non-ambiguously that distribution.
Conffile: pb
Example: osrelfile:
    fedora: /etc/fedora-release

osremovedotinver
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-Ver, os-Ver-arch).
Value: true|false. If true, then no '.' (dot) character is kept in the version name.
Conffile: pb
Example: osremovedotinver:
    redhat: true

osrepo
Nature: Optional (Mandatory if pbinstalltype is pkg)
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-Ver, os-Ver-arch).
Value: comma separated list of packages, yum repo or apt sources.list files to be added to the VE|VM to support package installation of pb
Conffile: pb
Example: osrepo:

osssha
Nature: Optional (Mandatory if rpm type of package)
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-Ver, os-Ver-arch).
Value: sha algorithm used by createrepo
Conffile: pb
Example: osssha:
    fedora-10: sha1

ossudoersmode
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-Ver, os-Ver-arch).
Value: suffix that will be used in the name of the packages created. By default, concatenation of OS name and version.
Conffile: pb
Example: ossudoersmode:
    novell: 640

ossuffix
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-Ver, os-Ver-arch).
Value: suffix that will be used in the name of the packages created. By default, concatenation of OS name and version.
Conffile: pb
Example: ossuffix:
    mandriva: mdv

ostype
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-Ver, os-Ver-arch).
Value: build type grouping packages family. This is used internally by pb to make
Conffile: pb
Example: ostype:
    rh: rpm, ostype md = rpm, ostype novell = rpm

osupd
### pb.upd

<table>
<thead>
<tr>
<th>Nature</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>OS (could be from the most generic up to the most specific from ostype, osfam, os, osver, osverarch)</td>
<td>OS command to launch in order to automatically update the VM/VE</td>
</tr>
<tr>
<td></td>
<td>Conffile</td>
<td>pb</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>fedora: sudo yum -y update</td>
</tr>
</tbody>
</table>

### ossueminorrel

<table>
<thead>
<tr>
<th>Nature</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>OS (could be from the most generic up to the most specific from ostype, osfam, os, osver, osverarch)</td>
<td>Should that OS distinguish between its minor version, considering them as different versions or not. Typically for CentOS where .x versions make incompatible changes.</td>
</tr>
<tr>
<td></td>
<td>Conffile</td>
<td>pb</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>centos: true</td>
</tr>
</tbody>
</table>

### pbadditionalgpg

<table>
<thead>
<tr>
<th>Nature</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>project (as defined in the -p option or pb environment variable)</td>
<td>GPG Key (hexadecimal) list of values separated by ',' that needs to be exported for this project.</td>
</tr>
<tr>
<td></td>
<td>Conffile</td>
<td>project</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>pb: 0x141B9FF237DB9883</td>
</tr>
</tbody>
</table>

### pbconfurl

<table>
<thead>
<tr>
<th>Nature</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>project (as defined in the -p option or pb environment variable)</td>
<td>B&lt;pb URL&gt; giving access to where the pb configuration dir is stored. Under that directory you have the same tree infrastructure as the upstream project, and under the pb packaging infra, including the project configuration file. Cf: man</td>
</tr>
<tr>
<td></td>
<td>Conffile</td>
<td>home</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>fossology: svn+ssh://user@svn.project-builder.org/mondo/svn/pb/projects</td>
</tr>
</tbody>
</table>

### pbdefdir

<table>
<thead>
<tr>
<th>Nature</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>project (as defined in the -p option or pb environment variable)</td>
<td>Local directory under which every pb related operation will take place. In this context, it is any shared directory that both the pb and the project can access.</td>
</tr>
<tr>
<td></td>
<td>Conffile</td>
<td>home</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>default: $ENV{'HOME'}/local/pb/projects</td>
</tr>
</tbody>
</table>

### pbgitremote

<table>
<thead>
<tr>
<th>Nature</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>project (as defined in the -p option or pb environment variable)</td>
<td>Remote name of the git repository used. The default is origin</td>
</tr>
<tr>
<td></td>
<td>Conffile</td>
<td>home</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>python-redfish: upstream</td>
</tr>
</tbody>
</table>

### pbgpgcheck

<table>
<thead>
<tr>
<th>Nature</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>project (as defined in the -p option or pb environment variable)</td>
<td>Whether the repository file should be generated specifying that gpg check is on. Note that signatures can fail making the repository file generated not work. The default is off.</td>
</tr>
<tr>
<td></td>
<td>Conffile</td>
<td>project</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>pbgpgcheck:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lintel: 0</td>
</tr>
</tbody>
</table>

### pbgpgserver

<table>
<thead>
<tr>
<th>Nature</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
</table>
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: The GPG server to use when looking for GPG keys.
Conffile: pb
Example: pbgpgserver:
    default: ipv4.pool.sks-keyservers.net

pbinstalltype
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: file or pkg. Indicates how pb will be installed during the setupve|vm phase in the virtual environment|machine, ... or upstream packages. Only the dev team needs to use file as packages do not yet exist for it. Or when no repository exists
Conffile: pb
Example: pbinstalltype:
    default: pkg

pblm
Nature: Optional (Mandatory if using announce command)
Key: project (as defined in the -p option or pb environment variable)
Value: white space separated list of e-mail addresses used to send announces with pb announce option.
Conffile: project
Example: pbml:
    pb: pb-announce@project-builder.org pb-devel@project-builder.org

pbpackager
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: Firstname Name and e-mail address of the person packaging the software.
Conffile: project
Example: pbpackager:
    pb: Bruno Cornec <bruno@project-builder.org>

pbparallel
Nature: Optional
Key: tool (pb or rpmbootstrap)
Value: number of processes to execute in parallel. By default use the number of available CPU cores
Conffile: pb
Example: pbparallel:
    pb: 12

pbpassfile
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: File containing the GPG passphrase that is used to sign packages
Conffile: home
Example: pbpassfile:
    pb: /users/me/secret/passfile

pbpasspath
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: The directory under which will be found your secret GPG key file.
Conffile: home
Example: pbpasspath:
    pb: /home/me/.gnupg

pbpassphrase
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: The GPG passphrase that is used to sign packages. Putting it in your conf
Conffile: home
Example: pbpassphrase:
    pb: TheSecretPassPhrase

pbpbr
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: whatever. As soon as this is defined, then that project is known as using pbr for source expansion.
Conffile: project
Example: pbpbr:
    python-redfish: 1

pbprojdir
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: local directory under which the project is locally exported. NB: a default value is
Conffile: home
Example: pbprojdir:
    mondorescue: $ENV{'HOME'}/local/mondorescue

pbrepo
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: URL of the server hosting the upstream tar file.
Conffile: project
Example: pbrepo:

pbshowsudo
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: false (by default), meaning that sudo commands executed with pb_system won't be shown in details, but that the...
Conffile: home
Example: pbshowsudo:
    mondorescue: true

pbssmtp
Nature: Optional (Mandatory if using the announce command)
Key: project (as defined in the -p option or pb environment variable)
Value: FQDN of the mail server to use to send announces. NB: a default value is
Conffile: home
Example: pbssmtp:
    mondorescue: localhost

pbsockscmd
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: name of the command to use to socksify network calls. NB: a default value is
Conffile: home
Example: pbsockscmd:
    default: tsocks
**pbstoponerr**

**Nature:** Optional

**Key:** project (as defined in the −p option or pb environment variable)

**Value:** false (by default), meaning that commands giving errors will not stop execution of the pb job.

**Conffile:** home

**Example:**

```
mondorescue: true
```

**pbprojurl**

**Nature:** Optional

**Key:** project (as defined in the −p option or pb environment variable)

**Value:** giving access to where the project is stored. Normally provided by the project, but could be overloaded for specific authentication information in the home configuration file or when using a DVCS

**Conffile:** home|project

**Example:**

```
linuxcoe: cvs+ssh://:ext:user@linuxcoe.cvs.sourceforge.net:/cvsroot/linuxcoe
```

**pbusesshagent**

**Nature:** Mandatory

**Key:** project (as defined in the −p option or pb environment variable)

**Value:** false means that you want pb to create a dedicated SSH key pair to dialog with VM|RM and false that you prefer to use an existing SSH Agent instead and existing keys

**Conffile:** pb

**Example:**

```
default: false
```

**pbwf**

**Nature:** Optional

**Key:** project (as defined in the −p option or pb environment variable)

**Value:** whatever. As soon as this is defined, then that project is known as not well formed (have a subdirectory in its root named project−version). This should be reported upstream to the project.

**Conffile:** project

**Example:**

```
afio: 1
```

**pkgtag**

**Nature:** Optional

**Key:** package (as provided in defpkgdir or extpkgdir)

**Value:** Tag that needs to be used in package name (on rpm: name-ver-tag.arch.rpm).

**Conffile:** project

**Example:**

```
mindi-busybox: 2
```

**pkgver**

**Nature:** Optional

**Key:** package (as provided in defpkgdir or extpkgdir)

**Value:** Version that needs to be used in package name (on rpm: name-ver-tag.arch.rpm)

**Conffile:** project

**Example:**

```
mindi-busybox: 1.7.3
```

**projcomponent**

**Nature:** Optional

**Key:** project (as defined in the −p option or pb environment variable)

**Value:** The component in the distribution repository, e.g. main for debian/ubuntu free software, (or non-free, contrib) or contrib for mandriva non core component e.g.

**Conffile:** project

**Example:**

```
Lintel: main
```

**proftag**
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: tag that will be used for all packages names (on rpm: name-ver-tag.arch.rpm)
Conffile: project
Example: projtag:
    mondorescue: 1

projver
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: version that will be used for all packages names (on rpm: name-ver-tag.arch.rpm)
Conffile: project
Example: projver:
    mondorescue: 2.2.9

rbsconf
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: for mock it's the configuration directory. For rinse it's its configuration
Conffile: ve
Example: rbsconf:
    default: /etc/mock, rbsconf default = /etc/pb/pb-rinse.conf

rbsb4pi
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: for mock it's not used. For rinse it's the script to call before doing installation (in order to change the mirror location).
Conffile: ve
Example: rbsb4pi:
    centos: /home/rinse/bin/before-post-install.sh

rbsmirrorsrv
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: URL for the mirror server for setting up a virtual environment
Conffile: ve
Example: rbsmirrorsrv:
    debian: http://mirrors1.kernel.org/

rbsmirrorupd
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch). The family name is generally used here.
Value: Relative path wrt rbsmirrorsrv where updates are located
Conffile: ve
Example: rbsmirrorupd:
    mageia: ../updates

rbsopt
Nature: Optional
Key: tool used for rpm based VE. Could be one of rpmbootstrap, rinse, mock, ...
Value: Additional option to pass to the command
Conffile: ve
Example: rbsopt:
    rpmbootstrap: -k

rbspi
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-verbatim).
Value: for mock it's not used. For rinse it's the script to call after doing installation for customization.
Conffile: ve
Example: rbspi:
   centos: /home/rinse/bin/post-install.sh

rmhost
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-verbatim).
Value: IP address or name of the Remote Machine running the OS mentioned in the key, accessed through ssh.
Conffile: rm
Example: rmhost:
   default: localhost - rmhost hpux-11.3-ia64 = 10.10.10.10 - rmhost mandriva-2010.2-x86_64

rmrmlist
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: list of comma separated OS (under the form of os-verbatim-arch). The corresponding machines running these distributions are given in the rmpool parameter.
Conffile: rm
Example: rmrmlist:
   default: mandriva-2010.2-i386,fedora-14-i386,rhel-6-i386,rhel-5-i386,ubuntu...

rmlogin
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: account name to use on the Remote Machine to build packages. Communication is done with ssh.
Conffile: rm
Example: rmlogin:
   default: pb

rmmonport
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: TCP port that is used to dialog with the monitor of the Remote Machine, to pass orders.
Conffile: rm
Example: rmmonport:
   default: 4444

rmntp
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: NTP server to contact for time accuracy with B<sopathcmd-ntpd> before building.
Conffile: rm
Example: rmntp:
   default: 1.pool.ntp.org

rmrmpath
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: path where to find configuration file for Remote Machines management.
Conffile: rm
Example: rmrmpath:
   default: /home/remote

rmrmpart
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch).
Value: Port number to use to communicate with the RM using the SSH protocol. This localport is redirected to the port 22 of the RM.
Conffile: rm
Example: rmpb: 2222, rmpb mondorescue = 2223

宿端口
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch).
Value: Time in seconds to wait before interacting with the RM. This may correspond to the time the RM takes to boot.
Conffile: rm
Example: rmtmout:
    default: 10

宿类型
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: ssh. For the moment, only ssh is supported as a communication means with the RM.
Conffile: rm
Example: rmrtype:
    default: ssh

宿目录
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: dirname into which packages are uploaded on the B<sshhost> machine.
Conffile: project
Example: sshdir:
    mondorescue: /pub/mondorescue

宿主机
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: hostname to connect to in order to deliver packages to the repository server.
Conffile: project
Example: sshhost:
    mondorescue: ftp.mondorescue.org

宿登录
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: login to use when connecting to the repository server B<sshhost> for package delivery. whoami result by default.
Conffile: project
Example: sshlogin:
    mondorescue: mylogin

宿端口
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: port to use when connecting to the repository server B<sshhost> for package delivery. 22 by default.
Conffile: project
Example: sshport:
    mondorescue: 22

宿文件
Not used yet.
testver
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: true (meaning this is a test version, whose tag will be generated automatically, based on 0+date in order to allow it to be inferior to tag 1 of the official version, and delivered under the test subdirectory of the ftp server).
Conffile: project
Example: testver:
    mondorescue: true

vedebtype
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: debootstrap (no other tool to create deb distro based chroot)
Conffile: ve
Example: vedebytype:
    default: debootstrap

velist
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: list of comma separated OS (under the form of os-ver-arch).
Conffile: ve
Example: velist:
    default: centos-4-i386,centos-5-i386,centos-4-x86_64,centos-5-x86_64,debian-5.0-i386,debian-5.0-x86_64,lsb-4.0.1-i386,lsb-4.0.1-x86_64

velogin
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: account name to use in the VE to build packages.
Conffile: ve
Example: velogin:
    default: pb

ventp
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: NTP server to contact for time accuracy with B<ospathcmd-ntpd> before building.
Conffile: ve
Example: ventp:
    default: 1.pool.ntp.org

vepath
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: path where to find VEs. Ve will be created and used under that path. For
Conffile: ve
Example: vePATH:
    default: /home/rpmbootstrap

verebuild
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: true|false. True means that the VE should be rebuild before usage.
Conffile: ve
Example: verebuild:
    default: true

verpmttype
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: rpmbootstrap|rinse|mock (different tools to create a chroot environment for
Conffile: ve
Example: verpmttype:
  default: rpmbootstrap

vesnap
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: true|false. True means that the snapshot of the VE should be used before
Conffile: ve
Example: vesnap:
  default: true

vetype
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: chroot|schroot|docker. There are two different ways of launching a Virtual
Conffile: ve
Example: vetype:
  default: chroot

vmbuildtm
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch).
Value: Time in seconds to wait before killing the VM if SSH port already used. This should correspond to the time to send files, build the project and get packages in the VM.
Conffile: project
Example: vmbuildtm:
  default: 600,vmbuildtm mandriva-2009.0-x86_64 = 1200

vmcmd
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch).
Value: Command to call to launch the VM emulator. It can contain some options. Another way to pass options to the VM manager command launcher is by using the PBVMOPT environment variable, which comes in addition to the option vmopt.
Conffile: vm
Example: vmcmd:
  default: /usr/bin/kvm

vmhost
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch).
Value: hostname to connect to in order to reach the VM through ssh. Generally redirected from a port on localhost.
Conffile: vm
Example: vmhost:
  default: localhost

vmlist
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: list of comma separated OS (under the form of os-ver-arch).
Conffile: vm
Example: vmlist:
  default: asianux-2-i386,asianux-3-i386,mandrake-10.1-i386,mandrake-10.2-x86_64

vmlogin
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: account name to use in the VM to build packages. Communication is done with ssh.
Conffile: vm
Example: vmlogin:
   default: pb

**vmmem**
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch).
Value: Memory size in MB to allocate to the VM.
Conffile: vm
Example: vmmem:
   default: 512

**vmmonport**
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: TCP port that is used to dialog with the monitor of the VM, to pass orders such as snapshot. Not really operational yet.
Conffile: vm
Example: vmmonport:
   default: 4444

**vmntp**
Nature: Optional
Key: project (as defined in the -p option or pb environment variable)
Value: NTP server to contact for time accuracy with B<ospathcmd-ntpdate> before building.
Conffile: vm
Example: vmntp:
   default: 1.pool.ntp.org

**vmopt**
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os-ver, os-ver-arch).
Value: list of options to pass to the VM manager command launcher for that distribution. Another way to pass options to the VM manager command launcher is by using the PBVMOPT environment variable, which comes in addition to this.
Conffile: vm
Example: vmopt:
   default: -m 384 -daemonize,vmopt mandriva-2009.0-1386 = -m 256 -daemonize

**vmpath**
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: path where to find VMs. They will be created and used under that path. For each VM os-ver-arch, it will create a os-ver-arch.qemu file below that point.
Conffile: vm
Example: vmpath:
   default: /home/qemu

**vmport**
Nature: Mandatory
Key: project (as defined in the -p option or pb environment variable)
Value: port number to use to communicate with the VM using the SSH protocol. This localport is redirected to the port 22 of the VM.
Conffile: vm
Example: vmport:
   pb: 2222,vmport mondorescue = 2223

**vmsize**
Nature: Mandatory
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch).
Value: Size of the VM to create when using the newvm command of pb.
Conffile: vm
Example: vmsize:
  default: 7G

vmsnap
Nature: Optional
Key: project (as defined in the −p option or pb environment variable)
Value: true|false. True means that the snapshot of the VM called pb should be used.
Conffile: vm
Example: vmsnap:
  default: true

vmtmout
Nature: Optional
Key: OS (could be from the most generic up to the most specific from ostype, osfamily, os, os−ver, os−ver−arch). NB: a default value is highly interesting here.
Value: Time in seconds to wait before interacting with the VM. This should correspond to the time the VM takes to boot.
Conffile: vm
Example: vmtmout:
  default: 180,vmtmout mandriva-2009.0-x86_64 = 500

vmttype
Nature: Mandatory
Key: project (as defined in the −p option or pb environment variable)
Value: qemu|kvm. For the moment, only QEMU or KVM are supported as virtualization technologies.
Conffile: vm
Example: vmttype:
  default: kvm

webdir
Nature: Optional
Key: project (as defined in the −p option or pb environment variable)
Value: Target directory containing the web content in the project that should be delivered when using the sbx|cms2webssh|pkg command of pb.
Conffile: project
Example: webdir:
  mondorescue: website

websshdir
Nature: Optional (when not using *2webssh commands)
Key: project (as defined in the −p option or pb environment variable)
Value: dirname into which content is uploaded on the B<websshhost> machine.
Conffile: project
Example: websshdir:
  mondorescue: /var/www/html

websshhost
Nature: Optional (when not using *2webssh commands)
Key: project (as defined in the −p option or pb environment variable)
Value: hostname to connect to in order to deliver content to the Web server.
Conffile: project
Example: websshhost:
  mondorescue: www.mondorescue.org

websshlogin
Nature: Optional (when not using *2webssh commands)
Key: project (as defined in the −p option or pb environment variable)
Value: login to use when connecting to the Web server B<websshhost> for content delivery.
Conffile: project
Example: websshlogin:
  mondorescue: mylogin

websshport
Nature: Optional (when not using *2webssh commands)
Key: project (as defined in the −p option or pb environment variable)
Value: port to use when connecting to the Web server B<websshhost> for content delivery.
Conffile: project
Example: websshport:
  mondorescue: 22

OTHER PARAMETERS

pb URLs
The pbprojurl and pbconfurl parameters support multiple schemas to point to the repositories to use.
They are parsed by project−builder.org to communicate with them.

The protocols can be git, git+svn, svk, svn, cvs, hg.
If you have write access to the repository, you'll generally use an ssh access with +ssh.

Examples:

pbprojurl:
  fossology: git+https://github.com/fossology/fossology.git
  linuxcoe: cvs+ssh://ext:bcornec@linuxcoe.cvs.sourceforge.net:/cvsroot/linuxcoe
  pb: svn+ssh://svn.mondorescue.org/prj/svn/pb

=> You access the upstream FOSSology project in read mode (by https) and the project uses git.
=> You access the upstream LinuxCOE project in write mode (by ssh) and the project uses CVS.
=> For Project−builder.org itself, you access the upstream LinuxCOE project in write mode (by ssh) and the project uses subversion.

=> You access the upstream project−builder.org project in write mode (by ssh) and the project uses subversion.

In some cases, there is no repository and the files are hosted remotely, in which case you use http(s) or ftp.

Examples:

pbprojurl:
  afio: ftp://localhost/src/afio−2.5.tar.gz

=> You access the project in read mode as a tar compressed file format using ftp.

pb Directories
In order to use project−builder.org to build packages for a project, you need to define a number of directories in your configuration file.

Tree will look like this:

maint pbdefdir PBDEFDIR
  dev dir (optional)
The first couple to declare is the pbconfurl and pbconfdir. They declare the location of the repository containing the configuration files and the local directory in which they are checked out. They do not need to be hosted in the upstream.

Examples:

**pbconfurl:**
- python-redfish: `git+ssh://git@github.com:bcornec/python-redfish.git`
- fossology: `git+https://github.com/fossology/fossology.git`
- uuwl: `git+svn+ssh://svn.mondorescue.org/prj/svn/pb/projects/uuwl/pbconf`
- pb: `git+svn+ssh://svn.mondorescue.org/prj/svn/pb/pbconf`
- afio: `git+svn+ssh://svn.project-builder.org/prj/svn/pb/projects/afio/pbconf`
- linuxcoe: `cvs+ssh://ext:bcornec@linuxcoe.cvs.sourceforge.net:/cvsroot/linuxcoe/pbconf`

**pbconfdir:**
- python-redfish: `$ENV{'HOME'}/Work/bruno/prj/python-redfish/pbconf`
- fossology: `$ENV{'HOME'}/Work/bruno/prj/fossology/git/pbconf`
- uuwl: `$ENV{'HOME'}/svn-git/pb/projects/uuwl/pbconf`

**pbdefdir:**
- default: `$ENV{'HOME'}/svn-git/pb/projects`
- python-redfish: `$ENV{'HOME'}/Work/bruno/prj`
- fossology: `$ENV{'HOME'}/Work/bruno/prj/fossology`
- uuwl: `$ENV{'HOME'}/svn-git`
- pb: `$ENV{'HOME'}/svn-git`
You access the project-builder.org configuration files for python-redfish in write mode (by ssh) and this project is using git as a VCS, that will also be used by project-builder.org to build packages. The first time, project-builder.org will clone the git repository from the upstream files for python-redfish in write mode (by ssh) and this project is using git as a VCS, that will also be used by project-builder.org to build packages. The first time, project-builder.org will clone the git repository from the upstream files for FOSSology in write mode (by ssh) and this project is using git as a VCS, that will also be used by project-builder.org to build packages. The first time, project-builder.org will clone the git repository from the upstream files for UUWL in write mode (by ssh) and this project is using SVN as a VCS to manage the sources. The first time, project-builder.org will clone the git repository from the upstream files for project-builder.org in write mode (by ssh) and this project is using SVN as a VCS to manage the sources. The first time, project-builder.org will clone the git repository from the upstream files for afio in write mode (by ssh) and this project is using SVN as a VCS to manage the sources. The first time, project-builder.org will clone the git repository from the upstream files for LinuxCOE in write mode (by ssh) and this project is using CVS as a VCS to manage the sources. The first time, project-builder.org will export the CVS repository.

Note that ultimately, if pbdefdir is not defined, project-builder.org will use /var/cache by default, which may fail if you do not have appropriate write rights.

Examples:

pbprojurl:
- python-redfish: git+ssh://git@github.com:bcornec/python-redfish.git
- fossology: git+ssh://git@github.com:fossology/fossology.git
- uuwl: git+svn+ssh://svn.mondorescue.org/prj/svn/uuwl
- pb: git+svn+ssh://svn.mondorescue.org/prj/svn/pb
- afio: ftp://localhost/src/afio-2.5.tar.gz
- linuxcoe: cvs+ssh://:ext:bcornec@linuxcoe.cvs.sourceforge.net:/cvsroot/linuxcoe

pbprojdir:
- fossology: $ENV{'HOME'}/Work/bruno/prj/fossology/git
- pb: $ENV{'HOME'}/svn-git/pb
- linuxcoe: $ENV{'HOME'}/LinuxCOE/cvs

pbdefdir:
- default: $ENV{'HOME'}/svn-git/pb/projects
- python-redfish: $ENV{'HOME'}/Work/bruno/prj
- fossology: $ENV{'HOME'}/Work/bruno/prj/fossology
- uuwl: $ENV{'HOME'}/svn-git
- pb: $ENV{'HOME'}/svn-git

In order to help you validate the value for a given parameter, you may want to use

```
pb −p pbproj getconf param_to_consider
```

### pb Environment Variables

The following environment variables are used by pb when declared to change its behavior:

**PBACCOUNT**
- Default: Empty
- Value: Login to use to connect to the VM/VE/RM. Also see −a option.

**pb**
- Default: Empty
- Value: Name of the project to build for. Also see −p option.

**PBROOTDIR**
- Default: Empty
- Value: Local directory in which to check out the repository. Also see −d option.
Default: Empty
Value: Root directory of the configuration files for this project. Also see −r option.

PBV
Default: Empty
Value: List of VM/VE/RM to build for, separated by ', '. Also see −m option.

PBVCSOPT
Default: Empty
Value: Options to pass to the VCS command when interacting with the repository.

PBVMOPT
Default: Empty
Value: Options to pass to the VM engine to launch VMs.

PBVMTMOUT
Default: 120
Value: Timeout in seconds to wait for the launch of the VM before communicating.

TMPDIR
Default: /tmp
Value: Directory where temporary files will be created.

ftp_proxy
http_proxy
https_proxy
Default: Empty
Value: URL of the proxy server to use for these protocols.

The following environment variables are generated by pb and can be used in build scripts:

PBBUILDDIR
Value: Build directory (pbbuild) where packages are created locally. See the schema of ProjectBuilder::Env man page.

PBCMSLOGFILE
Value: Intermediate log file generated for stable versions to create ChangeLog.

PBCONFDIR
Value: Configuration directory (pbconf) where configuration files for the project are stored. See the schema of ProjectBuilder::Env man page.

PBDEFDIR
Value: Default directory where the project-builder.org will host host files for the project.

PBDESTDIR
Value: Destination directory (pbdelivery) where intermediate tar files are created.

PBDIR
Value: Directory where an upstream version is located. See the schema of ProjectBuilder::Env man page.

PBETC
Value: .pbrc.yml configuration file of the user located in his HOME directory.
Value: E-mail address of the packager, used also to get GPG information. See pbpackager parameter.

PBPASSFILE
Value: File containing the pass phrase for the GPG signature. Used with PBPASSPATH.

PBPASSPATH
Value: Path of the file containing the pass phrase for the GPG signature. Used instead of PBPASSFILE.

PBPASSPHRASE
Value: Pass phrase for the GPG signature. Used instead of PBPASSPATH+PBPASSFILE.

website
Value: Name of the package built.

pbDIR
Value: Directory where an upstream project is located. See the schema of ProjectBuilder::Env man page. Correspond to pb dir under PBDEFDIR for project-builder.org conf files.

pbTAG
Value: Tag of the packages created, indicating the build procedure version. See pbprojtag parameter.

pbVER
Value: Version of the packages created. See pbprojver parameter.

2448MISION
Value: Revision of the project in the VCS. Revision for SVN, commit ID for git.

PBVMPORT
Value: Offset to the base port to communicate with the VM.

PBSOLDESTDIR
Value: Target directory for the Solaris prototype.

COPYRIGHT
(c) B. Cornec 2007−today
Eric Anderson's changes are (c) Copyright 2012 Hewlett Packard
Provided under the Attribution−ShareAlike 3.0 Unported (CC BY−SA 3.0) Creative Common License