

`/usr/share/doc/${PF}/html.docompress path ...` adds paths to the inclusion list for compression. `docompress -x path ...` adds paths to the exclusion list. See CONTROLLABLE-COMPRESS on page 60.

**dodoc recursion** If the `-r` switch is given as first argument and followed by directories, files from there are installed recursively. See DODOC on page 56.

**doins symlink support** Symbolic links are now properly installed when using recursion (`-r` switch). See DOINS on page 57.

**nonfatal for commands** If you call `nonfatal` the command given as argument will not abort the build process in case of a failure (as is the default) but will return non-zero on failure rather than aborting the build. See NONFATAL on page 53.

**PROPERTIES** Is mandatory for all package managers now to support interactive installs.

#### REPLACING\_VERSIONS, REPLACED\_BY\_VERSION

These variables, valid in `pkg_*`, contain a list of all versions (PVR) of this package that we are replacing, and the version that is replacing the current one, respectively. See REPLACE-VERSION-VARS on page 51.

**MERGE\_TYPE** This variable contains one of three possible values to allow checks if it is normal merge with compilation and installation (`source`), installation of a binary package (`binary`), or a compilation without installation (`buildonly`). See MERGE-TYPE on page 48.

**REQUIRED\_USE** This variable can be used similar to the (R|P)DEPEND variables and define sets of USE flag combinations that are not allowed. All elements can be further nested to achieve more functionality.

**Forbidden combination** To prevent activation of `flag1` if `flag2` is enabled use `"flag2? ( !flag1 )"`.

**OR** If at least one USE flag out of many must be activated on `flag1` use `"flag1? ( || ( flag2 flag3 ... ) )"`.

**XOR** To allow exactly one USE flag out of many use `"^( flag1 flag2 ... )"`.

See REQUIRED-USE on page 26.

## Removals/Bans

**dohard, dosed** Both functions are not allowed any more. See BANNED-COMMANDS on page 53.

**No RDEPEND fall-back** The package manager will not fall back to `RDEPEND=DEPEND` if `RDEPEND` is undefined. See RDEPEND-DEPEND on page 27.

**S fallback changes** The value of the variable `S` will not automatically be changed to `WORKDIR`, if `S` is not a directory, but abort. Virtual packages are the only exception. See S-WORKDIR-FALLBACK on page 35.

**AA, KV** These variables are not defined any more. See AA on page 45 and KV on page 48.

# EAPI Cheat Sheet

Gentoo PMS team\*

Version 4.0  
14th July 2011

## Abstract

An overview of the main EAPI changes in Gentoo, for ebuild authors. For full details, consult the Package Manager Specification found on the project page; this is an incomplete summary only.

Official Gentoo EAPIs are consecutively numbered integers (0, 1, 2, ...). Except where otherwise noted, an EAPI is the same as the previous EAPI. All labels refer to the PMS document itself, built from the same checkout as this overview.

Please report mistakes in or enhancements to this document via the Gentoo bug tracking system<sup>1</sup> to the original author or the PMS team.

This document is released under the Creative Commons Attribution-Share Alike 3.0 Licence<sup>2</sup>.

## EAPI 0

If there is no EAPI explicitly specified, EAPI 0 is assumed. Currently there is no full specification what EAPI 0 includes. Portage, official ebuild documentation and existing ebuilds set the standard. If you think you found a bug, you should file a bug report nonetheless.

\*<http://www.gentoo.org/proj/en/qa/pms.xml>

<sup>1</sup><http://bugs.gentoo.org/>

<sup>2</sup><http://creativecommons.org/licenses/by-sa/3.0/>

## EAPI 1

### Additions/Changes

**USE defaults** A USE flag can be marked as mandatory (if not disabled explicitly by user configuration) with a + sign in front. See USE-DEFAULTS on page [25](#).

**Named slot dependencies** Dependencies can explicitly request a specific slot by using the `dev-libs/foo: SLOT_name` syntax. See SLOT-DEFS on page [32](#).

## EAPI 2 (2008-09-25)

### Additions/Changes

**SRC\_URI arrows** Allows redirection of upstream file naming scheme. By using `SRC_URI="http://some/url -> foo" the file is saved as foo in DISTDIR`. See SRC-URI-ARROWS on page [34](#).

**USE dependencies** Dependencies can specify USE flag requirements on their target, removing the need for `build_with_use` checks. A more powerful syntax that does not require the flag to be in IUSE is in EAPI 4. **[opt]** The flag must be enabled.

**[opt=]** The flag must be enabled if the flag is enabled for the package with the dependency, or enabled otherwise.

**[opt?]** The flag must be enabled if the flag is enabled for the package with the dependency.

**[opt?]** The flag must be disabled if the use flag is disabled for the package with the dependency.

**[opt]** The flag must be disabled.

See USE-DEFS on page [31](#).

**Blocker syntax** A single exclamation mark as a blocker may be ignored by the package manager as long as the stated package is uninstalled later on. Two exclamation

marks are a strong blocker and will always be respected. See BANG-STRENGTH on page [32](#).

**src\_configure, src\_prepare** Both new phases provide finer granularity in the ebuild's structure. Configure calls should be moved from `src_compile` to `src_configure`. Patching and similar preparation must now be done in `src_prepare`, not `src_unpack`. See SRC-PREPARE on page [36](#) and SRC-CONFIGURE on page [37](#).

**Default phase functions** The default functions for the phases `pkg_nofetch`, `src_unpack`, `src_prepare`, `src_configure`, `src_compile` and `src_test` can be called via `default_phase_name`, so duplicating the standard implementation is no longer necessary for small additions. The short-hand `default` function calls the current phase's `default` function automatically, so any small additions you need will not be accompanied by a complete reimplementaton of the phase. See DEFAULT-PHASE-FUNCS on page [40](#) and DEFAULT-FUNCS on page [63](#).

**domain language support** The domain installation function recognizes language specific man page extensions and behaves accordingly. This behaviour can be inhibited by the `-il8n` switch with EAPI 4. See DOMAIN-LANGS on page [58](#).

## EAPI 3 (2010-01-18)

### Additions/Changes

**Support for .xz** Unpack of .xz and .tar.xz files is possible without any custom `src_unpack` functions. See UNPACK-EXTENSIONS on page [62](#).

**Offset prefix** Supporting installation on Prefix-enabled systems<sup>3</sup> will be easier with this EAPI.

## EAPI 4 (2011-01-17)

### Additions/Changes

**pkg\_prepend** Some useful checks (kernel options for example) can be placed in this new phase to inform the user early (when just pretending to emerge the package). Most checks should usually be repeated in `pkg_setup`. See PKG-PREPEND on page [35](#).

**src\_install** The `src_install` phase is no longer empty but has a default now. This comes along with an accompanying `default` function. See SRC-INSTALL-4 on page [38](#).

**pkg\_info on non-installed packages** The `pkg_info` phase can be called even for non-installed packages. Be warned that dependencies might not have been installed at execution time. See PKG-INFO on page [40](#).

**econft changes** The helper function now always activates `--disable-dependency-tracking`. See ECONFOPTIONS on page [55](#).

**USE dependency defaults** In addition to the features offered in EAPI 2 for USE dependencies, a (+) or (-) can be added after a USE flag (mind the parentheses). The former specifies that flags not in IUSE should be treated as enabled; the latter, disabled. Cannot be used with USE\_EXPAND flags. This mimics parts of the behaviour of `--missing` in `build_with_use`. See USE-DEF-DEFAULTS on page [33](#).

**Controllable compression** All items in the `doc`, `info`, `man` subdirectories of `/usr/share/` may be compressed on-disk after `src_install`, except for

<sup>3</sup><http://prefix.gentoo.org/>