EAPI Cheat Sheet

Christian Faulhammer
fauli@gentoo.org

Ulrich Müller
ulm@gentoo.org

Version 7.0
30th April 2018

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;[1] 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.

EAPIs 0, 1, 2, 3, and 4

Omitted for lack of space. See version 5.0 of this document for differences between these previous EAPIs.

EAPI 5 (2012-09-20)

Additions/Changes

Sub-slots The `SLOT` variable and slot dependencies may contain an optional sub-slot part that follows the regular slot, delimited by a `/` character; for example `2/2.30`. The sub-slot is used to represent cases in which an upgrade to a new version of a package with a different sub-slot may require dependent packages to be rebuilt. If the

[1] https://wiki.gentoo.org/wiki/Project:Package_Manager_Specification

[2] http://creativecommons.org/licenses/by-sa/3.0/

sub-slot is not specified in `SLOT`, it defaults to the regular slot. See SUB-SLOT on page 38.

Slot operator dependencies Package dependencies can specify one of the following operators as a suffix, which will affect updates of runtime dependencies:

`:*` Any slot value is acceptable. The package will not break when its dependency is updated.

`:=` Any slot value is acceptable, but the package can break when its dependency is updated to a different slot (or sub-slot).

See SLOT-OPERATOR-DEPS on page 38.

Profile `IUSE` injection Apart from the USE flags explicitly listed in `IUSE`, additional flags can be implicitly provided by profiles. See PROFILE-IUSE-INJECT on page 57.

At-most-one-of groups In `REQUIRED_USE` you can use "`?? ( flag1 flag2 ... )`" to allow zero or one USE flag out of many. See AT-MOST-ONE-OF on page 35.

Parallel tests The default for `src_test` runs `emake` without `-j1` now. See PARALLEL-TESTS on page 44.

`econf` changes The `econf` function now always passes `--disable-silent-rules` to `configure`. See ECONF-OPTIONS on page 65.

`has_version` and `best_version` changes The two helpers support a `--host-root` option that causes the query to apply to the host root instead of `ROOT`. See PM-QUERY-OPTIONS on page 62.

`usex` Usage for this helper function is `usex` *<USE flag> [true1] [false1] [true2] [false2]*. If the USE flag is set, outputs *[true1][true2]* (defaults to `yes`), otherwise outputs *[false1][false2]* (defaults to `no`). See USEX on page 72.

`doheader` and `newheader` These new helper functions install the given header file(s) into `/usr/include`. The `-r` option enables recursion for `doheader`, similar to `doins`. See DOHEADER on page 67.

`new*` standard input The `newins` etc. commands read from standard input if the first argument is `-` (a hyphen). See NEWFOO-STDIN on page 69.

`EBUILD_PHASE_FUNC` This variable is very similar to `EBUILD_PHASE`, but contains the name of the current ebuild function. See EBUILD-PHASE-FUNC on page 55.

Stable use masking/forcing New files `use.stable.{mask,force}` and `package.use.stable.{mask,force}` are supported in profile directories. They are similar to their non-`stable` counterparts, but act only on packages that would be merged due to a stable keyword. See STABLEMASK on page 25.

EAPI 6 (2015-11-13)

Additions/Changes

Bash version Ebuilds can use features of Bash version 4.2 (was 3.2 before). See BASH-VERSION on page 29.

`failglob` The `failglob` option of Bash is set in global scope, so that unintentional pattern expansion will be caught as an error. See FAILGLOB on page 29.

Locale settings It is ensured that the behaviour of case modification and collation order for ASCII characters (`LC_CTYPE` and `LC_COLLATE`) are the same as in the POSIX locale. See LOCALE-SETTINGS on page 56.

`src_prepare` This phase function has a default now, which applies patches from the `PATCHES` variable with the new `eapply` command, and user-provided patches with `eapply_user`. See SRC-PREPARE-6 on page 43.

`src_install` The default implementation uses the new `einstalldocs` function for installing documentation. See SRC-INSTALL-6 on page 45.

`nonfatal die` When `die` or `assert` are called under the `nonfatal` command and with the `-n` option, they will not abort the build process but return with an error. See NONFATAL-DIE on page 63.

`unpack` changes `unpack` has been extended:

- **Pathnames** Both absolute paths and paths relative to the working directory are accepted as arguments. See UNPACK-ABSOLUTE on page 75.
- **`.txz` files** Suffix `.txz` for xz compressed tarballs is recognised. See UNPACK-EXTENSIONS on page 75.
- **Filename case** Character case of filename extensions is ignored. See UNPACK-IGNORE-CASE on page 75.

`econf` changes Options `--docdir` and `--htmldir` are passed to `configure`, in addition to the existing options. See ECONF-OPTIONS on page 65.

`eapply` The `eapply` command is a simplified substitute for `epatch`, implemented in the package manager. The patches from its file or directory arguments are applied using `patch -p1`. See EAPPLY on page 63.

`eapply_user` The `eapply_user` command permits the package manager to apply user-provided patches. It must be called from every `src_prepare` function. See EAPPLY-USER on page 63.

`einstalldocs` The `einstalldocs` function will install the files specified by the `DOCS` variable (or a default set of files if `DOCS` is unset) and by the `HTML_DOCS` variable. See EINSTALLDOCS on page 76.

`in_iuse` The `in_iuse` function returns true if the USE flag given as its argument is available in the ebuild for USE queries. See IN-IUSE on page 73.

`get_libdir` The `get_libdir` command outputs the `lib*` directory basename suitable for the current ABI. See GET-LIBDIR on page 76.

Removals/Bans

`einstall` No longer allowed. Use `emake install` as replacement. See BANNED-COMMANDS on page 61.

EAPI 7 (2018-04-30)

Additions/Changes

`package.*` and `use.*` These profile files can be directories instead of regular files. This is intended to be used in overlays only. See PACKAGE-MASK-DIR on page 19 and PROFILE-FILE-DIRS on page 24.

`||` and `^^` dependency groups These groups now evaluate to false when they are empty (for example, if there are only unmatched use dependencies inside of them). See EMPTY-DEP-GROUPS on page 36.

No trailing slash The paths specified by `ROOT`, `EROOT`, `D`, and `ED` no longer end with a slash. Thus, default `ROOT` is empty now. See TRAILING-SLASH on page 58.

Cross compilation support Several variables have been added and some commands have been extended for better cross compilation support:

- **`BDEPEND`** Build dependencies are divided into two classes: `BDEPEND` for native build tools (`CBUILD`); `DEPEND` for dependencies compatible with the system being built (`CHOST`). See BDEPEND on page 34.
- **`SYSROOT`** The path to the root directory for `DEPEND` type dependencies. See SYSROOT on page 53.
- **`ESYSROOT`** The concatenation of the `SYSROOT` and `EPREFIX` paths, for convenience.
- **`BROOT`** The prefixed root directory path for `BDEPEND` type dependencies, typically executable build tools. See BROOT on page 53.
- **`econf`** Option `--with-sysroot=${ESYSROOT}` is passed to configure, if this option is supported. See ECONF-OPTIONS on page 65.
- **`has_version` and `best_version`** These helpers support `-b`, `-d` or `-r` options, causing the query to apply to `BDEPEND`, `DEPEND` or `RDEPEND` (the default). This replaces the `--host-root` option. See PM-QUERY-OPTIONS on page 62.

Environment blacklist Any environment variable listed in the profile-defined `ENV_UNSET` variable will be unset by the package manager. See ENV-UNSET on page 56.

`patch` All inputs valid for GNU patch version 2.7 are supported. Especially, this includes support for git-formatted patches. See GNU-PATCH on page 60.

`nonfatal` In addition to its definition as a shell function, the `nonfatal` wrapper has now a fallback implementation as an external command. Thus, it can be called from other commands. See NONFATAL on page 61.

Output commands `einfo` and friends no longer use stdout, so inside of command substitution their output won't be caught. See OUTPUT-NO-STDOUT on page 62.

`eqawarn` The `eqawarn` output command is supported in the package manager itself. See EQAWARN on page 63.

`die` in subshell The `die` command is guaranteed to work in a subshell context. See SUBSHELL-DIE on page 63.

domo destination `domo` installs the specified files under `/usr/share/locale` instead of `${DESTTREE}/share/locale`. See DOMO-PATH on page 68.

Controllable stripping The `dostrip -x` command can be used to add paths to an exclusion list for stripping of debug symbols, to allow more fine-grained control than with `RESTRICT="strip"`. See DOSTRIP on page 72.

Version manipulation and comparison commands

`ver_cut` *range* [*version*] Print the version substring specified by *range*. *version* defaults to `PV`.

`ver_rs` *range repl* . . . [*version*] Replace all version separators in *range* by string *repl*. Multiple *range repl* pairs are allowed. *version* defaults to `PV`.

`ver_test` [*v1*] *op v2* Check if the relation *v1 op v2* is true. *v1* defaults to `PVR`; *op* can be `-eq`, `-ne`, `-gt`, `-ge`, `-lt` or `-le`.

See VER-COMMANDS on page 73.

Removals/Bans

`package.provided` Deprecated since a long time and finally dropped. See PACKAGE-PROVIDED on page 25.

`PORTDIR` and `ECLASSDIR` No longer defined, because ebuilds should not directly access files in the repository. See PORTDIR on page 53 and ECLASSDIR on page 53.

`DESTTREE` and `INSDESTTREE` Not defined any more. Use the `into` and `insinto` commands instead. See DESTTREE on page 54 and INSDESTTREE on page 54.

`dohtml` No longer allowed. `doins -r` can be used as a replacement. See BANNED-COMMANDS on page 61.

`dolib` and `libopts` No longer allowed. The specific `dolib.a` or `dolib.so` commands should be used as replacement. See BANNED-COMMANDS on page 61.