

NAME

Dpkg::Deps::KnownFacts – list of installed real and virtual packages

DESCRIPTION

This object represents a list of installed packages and a list of virtual packages provided (by the set of installed packages).

METHODS

`$facts = Dpkg::Deps::KnownFacts->new();`

Creates a new object.

`$facts->add_installed_package($package, $version, $arch, $multiarch)`

Records that the given version of the package is installed. If `$version/$arch` is undefined we know that the package is installed but we don't know which version/architecture it is. `$multiarch` is the Multi-Arch field of the package. If `$multiarch` is undef, it will be equivalent to "Multi-Arch: no".

Note that `$multiarch` is only used if `$arch` is provided.

`$facts->add_provided_package($virtual, $relation, $version, $by)`

Records that the "\$by" package provides the `$virtual` package. `$relation` and `$version` correspond to the associated relation given in the Provides field (if present).

`($check, $param) = $facts->check_package($package)`

`$check` is one when the package is found. For a real package, `$param` contains the version. For a virtual package, `$param` contains an array reference containing the list of packages that provide it (each package is listed as [`$provider, $relation, $version`]).

This function is obsolete and should not be used. `Dpkg::Deps::KnownFacts` is only meant to be filled with data and then passed to `Dpkg::Deps` methods where appropriate, but it should not be directly queried.

`$facts->evaluate_simple_dep()`

This method is private and should not be used except from within `Dpkg::Deps`.

CHANGES**Version 1.01 (dpkg 1.16.1)**

New option: `Dpkg::Deps::KnownFacts->add_installed_package()` now accepts 2 supplementary parameters (`$arch` and `$multiarch`).

Deprecated method: `Dpkg::Deps::KnownFacts->check_package()` is obsolete, it should not have been part of the public API.

Version 1.00 (dpkg 1.15.6)

Mark the module as public.