

**NAME**

Dpkg::Deps::Multiple – base module to represent multiple dependencies

**DESCRIPTION**

The Dpkg::Deps::Multiple module provides objects implementing various types of dependencies. It is the base class for Dpkg::Deps::{AND,OR,Union}.

**METHODS**

`$dep = Dpkg::Deps::Multiple->new(%opts);`

Creates a new object.

`$dep->reset()`

Clears any dependency information stored in `$dep` so that `$dep->is_empty()` returns true.

`$dep->add(@deps)`

Adds new dependency objects at the end of the list.

`$dep->get_deps()`

Returns a list of sub-dependencies.

`$dep->sort()`

Sorts alphabetically the internal list of dependencies.

`$dep->arch_is_concerned($arch)`

Returns true if at least one of the sub-dependencies apply to this architecture.

`$dep->reduce_arch($arch)`

Simplifies the dependencies to contain only information relevant to the given architecture. The non-relevant sub-dependencies are simply removed.

This trims off the architecture restriction list of Dpkg::Deps::Simple objects.

`$dep->has_arch_restriction()`

Returns the list of package names that have such a restriction.

`$dep->profile_is_concerned()`

Returns true if at least one of the sub-dependencies apply to this profile.

`$dep->reduce_profiles()`

Simplifies the dependencies to contain only information relevant to the given profile. The non-relevant sub-dependencies are simply removed.

This trims off the profile restriction list of Dpkg::Deps::Simple objects.

`$dep->is_empty()`

Returns true if the dependency is empty and doesn't contain any useful information. This is true when a (descendant of) Dpkg::Deps::Multiple contains an empty list of dependencies.

`$dep->merge_union($other_dep)`

This method is not meaningful for this object, and will always croak.

**CHANGES****Version 1.02 (dpkg 1.17.10)**

New methods: Add `$dep->profile_is_concerned()` and `$dep->reduce_profiles()`.

**Version 1.01 (dpkg 1.16.1)**

New method: Add `$dep->reset()`.

**Version 1.00 (dpkg 1.15.6)**

Mark the module as public.