NAME

Lintian::Processable −− An (abstract) object that Lintian can process

SYNOPSIS

```
 use Lintian::Processable::Package;

 # Instantiate via Lintian::Processable::Package
 my $proc = Lintian::Processable::Package->new ('lintian_2.5.0_all.deb');
 my $pkg_name = $proc->pkg_name;
 my $pkg_version = $proc->pkg_version;
 # etc.
```

DESCRIPTION

Instances of this perl class are objects that Lintian can process (e.g. deb files). Multiple objects can then be combined into groups, which Lintian will process together.

CLASS METHODS

new_from_metadata (TYPE, PARAGRAPH[, BASEPATH])

Returns a Lintian::Processable from a PARAGRAPH in a Sources or a Packages file with the following exception.

If the PARAGRAPH has a field named "pkg_path", then that is used instead of creating the path from BASEPATH path concatenated with the TYPE specific field(s). Hench BASEPATH is optional if and only if, the paragraph has a field called "pkg_path".

The TYPE parameter determines the type of the processable and is required.

NB: Optional fields (e.g. "Source" for binaries) may be omitted in PARAGRAPH as usual. In this case, the respective values are computed from the required fields according to the Policy Manual.

INSTANCE METHODS

$proc−>pkg_name

Returns the package name.

$proc−>pkg_version

Returns the version of the package.

$proc−>pkg_path

Returns the path to the packaged version of actual package. This path is used in case the data needs to be extracted from the package.

Note: This may return the path to a symlink to the package.

$proc−>pkg_type

Returns the type of package (e.g. binary, source, udeb ...)

$proc−>pkg_arch

Returns the architecture(s) of the package. May return multiple values from changes processables. For source processables it is "source".

$proc−>pkg_src

Returns the name of the source package.

$proc−>pkg_src_version

Returns the version of the source package.

$proc−>tainted

Returns a truth value if one or more fields in this Processable is tainted. On a best effort basis tainted fields will be sanitized to less dangerous (but possibly invalid) values.

$proc−>identifier

Produces an identifier for this processable. The identifier is based on the type, name, version and architecture of the package.

`$proc`−>group([$group])

Returns the group `$proc` is in, if any. If the processable is not in a group, this returns `undef`.

Can also be used to set the group of this processable.

`$proc`−>info

Returns `$info` element for this processable.

Note: This method must be implemented by sub-classes unless they provide an ‘‘info’’ field.

`$proc`−>clear_cache

Discard the info element, so the memory used by it can be reclaimed. Mostly useful when checking a lot of packages (e.g. on lintian.d.o).

Note: By default this does nothing, but it may (and should) be overridden by sub-classes.

`$proc`−>get_field ($field[, `$def`])

Optional method to access a field in the underlying data set.

Returns `$def` if the field is not present or the implementation does not have (or want to expose) it. This method is *not* guaranteed to return the same value as ‘‘$proc−>info−>field ($field, `$def`)’’.

If `$def` is omitted is defaults to `undef`.

Default implementation accesses them via the hashref stored in ‘‘extra-fields’’ if present. If the field is present, but not defined `$def` is returned instead.

NB: This is mostly an optimization used by Lintian::Lab to avoid (re−)reading the underlying package data.

AUTHOR

Originally written by Niels Thykier <niels@thykier.net> for Lintian.

SEE ALSO

lintian (1)

Lintian::ProcessableGroup