

NAME

Lintian::Group -- A group of objects that Lintian can process

SYNOPSIS

```
use Lintian::Group;

my $group = Lintian::Group->new('lintian_2.5.0_i386.changes');
```

DESCRIPTION

Instances of this perl class are sets of processables. It allows at most one source and one changes or buildinfo package per set, but multiple binary packages (provided that the binary is not already in the set).

METHODS

`$group->pooldir`
Returns or sets the pool directory used by this group.

`$group->name`
Returns a unique identifier for the group based on source and version.

`$group->binary`
Returns a hash reference to the binary processables in this group.

`$group->buildinfo`
Returns the buildinfo processable in this group.

`$group->changes`
Returns the changes processable in this group.

`$group->source`
Returns the source processable in this group.

`$group->udeb`
Returns a hash reference to the udeb processables in this group.

`jobs`
Returns or sets the max number of jobs to be processed in parallel.
If the limit is 0, then there is no limit for the number of parallel jobs.

`cache`
Cache for some items.

`profile`
Hash with active jobs.

`saved_direct_dependencies`
`saved_direct_reliants`
`saved_spelling_exceptions`
`shared_storage`

`Lintian::Group->init_from_file (FILE)`
Add all processables from .changes or .buildinfo file FILE.

`unpack`
Unpack this group.

`process`
Process group.

`clean_lab`
Removes the lab files to conserve disk space. Global destruction will also get these unless we are keeping the lab.

`$group->add_processable($proc)`
Adds `$proc` to `$group`. At most one source and one changes `$proc` can be in a `$group`. There can be multiple binary `$proc`'s, as long as they are all unique. Successive buildinfo `$proc`'s are

silently ignored.

This will error out if an additional source or changes `$proc` is added to the group. Otherwise it will return a truth value if `$proc` was added.

`$group->get_processables([$type])`

Returns an array of all processables in `$group`. The processables are returned in the following order: changes (if any), source (if any), all binaries (if any) and all udebs (if any).

This order is based on the original order that Lintian processed packages in and some parts of the code relies on this order.

Note if `$type` is given, then only processables of that type is returned.

`$group->get_binary_processables`

Returns all binary (and udeb) processables in `$group`.

If `$group` does not have any binary processables then an empty list is returned.

`$group->info`

`direct_dependencies (PROC)`

If PROC is a part of the underlying processable group, this method returns a listref containing all the direct dependencies of PROC. If PROC is not a part of the group, this returns undef.

Note: Only strong dependencies (Pre-Depends and Depends) are considered.

Note: Self-dependencies (if any) are *not* included in the result.

`direct_reliants (PROC)`

If PROC is a part of the underlying processable group, this method returns a listref containing all the packages in the group that rely on PROC. If PROC is not a part of the group, this returns undef.

Note: Only strong dependencies (Pre-Depends and Depends) are considered.

Note: Self-dependencies (if any) are *not* included in the result.

`$ginfo->type`

Return the type of this collect object (which is the string 'group').

`spelling_exceptions`

Returns a hashref of words, which the spell checker should ignore. These words are generally based on the package names in the group to avoid false-positive "spelling error" when packages have "fun" names.

Example: Package alot-doc (#687464)

`$group->clear_cache`

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

AUTHOR

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

SEE ALSO

lintian (1)

Lintian::Processable