NAME

Lintian::Collect – Lintian interface to package data collection

SYNOPSIS

```
my ($name, $type, $dir) = ('foobar', 'udeb', '/some/abs/path');
my $collect = Lintian::Collect->new ($name, $type, $dir);
$name = $collect->name;
$type = $collect->type;
```

DESCRIPTION

Lintian::Collect provides the shared interface to package data used by source, binary and udeb packages and .changes files. It creates an object of the appropriate type and provides common functions used by the collection interface to all types of package.

Usually instances should not be created directly (exceptions include collections), but instead be requested via the info method in Lintian::Lab::Entry.

This module is in its infancy. Most of Lintian still reads all data from files in the laboratory whenever that data is needed and generates that data via collect scripts. The goal is to eventually access all data via this module and its subclasses so that the module can cache data where appropriate and possibly retire collect scripts in favor of caching that data in memory.

CLASS METHODS

new (PACKAGE, TYPE, BASEDIR[, FIELDS]))

Creates a new object appropriate to the package type. TYPE can be retrieved later with the ''type'' method. Croaks if given an unknown TYPE.

PACKAGE is the name of the package and is stored in the collect object. It can be retrieved with the ''name'' method.

BASEDIR is the base directory for the data and should be absolute.

If FIELDS is given it is assumed to be the fields from the underlying control file. This is only used to avoid an unnecessary read operation (possibly incl. an ar | gzip pipeline) when the fields are already known.

INSTANCE METHODS

In addition to the instance methods documented here, see the documentation of Lintian::Collect::Source, Lintian::Collect::Binary and Lintian::Collect::Changes for instance methods specific to source and binary / udeb packages and .changes files.

name

Returns the name of the package.

Needs-Info requirements for using *name*: none

type

Returns the type of the package.

Needs-Info requirements for using *type*: none

base_dir

Returns the base_dir where all the package information is stored.

Needs-Info requirements for using *base_dir*: none

lab_data_path ([ENTRY])

Return the path to the ENTRY in the lab. This is a convenience method around base_dir. If ENTRY is not given, this method behaves like base_dir.

Needs-Info requirements for using *lab_data_path*: ''base_dir''

unfolded_field (FIELD)

This method returns the unfolded value of the control field FIELD in the control file for the package. For a source package, this is the *.dsc file; for a binary package, this is the control file in the control

section of the package.

If FIELD is passed but not present, then this method returns undef.

Needs-Info requirements for using *unfolded_field*: none

field ([FIELD[, DEFAULT]])
: If FIELD is given, this method returns the value of the control field FIELD in the control file for the package. For a source package, this is the *.dsc file; for a binary package, this is the control file in the control section of the package.

  If FIELD is passed but not present, then this method will return DEFAULT (if given) or undef.

  Otherwise this will return a hash of fields, where the key is the field name (in all lowercase).

  Needs-Info requirements for using *field*: none

is_non_free
: Returns a truth value if the package appears to be non-free (based on the section field; ''non−free/*'' and ''restricted/*'')

  Needs-Info requirements for using *is_non_free*: ''field ([FIELD[, DEFAULT]])''

AUTHOR

Originally written by Russ Allbery <rra@debian.org> for Lintian.

SEE ALSO

lintian (1), Lintian::Collect::Binary, Lintian::Collect::Changes, Lintian::Collect::Source