NAME

lintian − Static analysis tool for Debian packages

SYNOPSIS

lintian [*action*] [*options*] [*packages*] ...

DESCRIPTION

Lintian dissects Debian packages and reports bugs and policy violations. It contains automated checks for many aspects of Debian policy as well as some checks for common errors.

There are two ways to specify binary, udeb or source packages for Lintian to process: by file name (the .deb file for a binary package or the .dsc file for a source package), or by naming a *.changes* file.

If you specify a *.changes* file, Lintian will process all packages listed in that file. This is convenient when checking a new package before uploading it.

If you specify packages to be checked or use the option **−−packages−from−file**, the packages requested will be processed. Otherwise, if *debian/changelog* exists, it is parsed to determine the name of the .changes file to look for in the parent directory (when using the actions **−−check** or **−−unpack**). See ‘‘CHECKING LAST BUILD’’ for more information.

OPTIONS

Actions of the lintian command: (Only one action can be specified per invocation)

−c, **−−check**
Run all checks over the specified packages. This is the default action.

−C chk1,chk2,..., **−−check−part** chk1,chk2,...
Run only the specified checks. You can either specify the name of the check script or the abbreviation. For details, see the ‘‘CHECKS’’ section below.

−F, **−−ftp−master−rejects**
Run only the checks that issue tags that result in automatic rejects from the Debian upload queue. The list of such tags is refreshed with each Lintian release, so may be slightly out of date if it has changed recently.

This is implemented via a profile and thus this option cannot be used together with **−−profile**.

−T tag1,tag2,..., **−−tags** tag1,tag2,...
Run only the checks that issue the requested tags. The tests for other tags within the check scripts will be run but the tags will not be issued.

With this options all tags listed will be displayed regardless of the display settings.

−−tags−from−file filename
Same functionality as **−−tags**, but read the list of tags from a file. Blank lines and lines beginning with # are ignored. All other lines are taken to be tag names or comma-separated lists of tag names to (potentially) issue.

With this options all tags listed will be displayed regardless of the display settings.

−u, **−−unpack**
Unpacks the package will all collections. See the ‘‘COLLECTION’’ section below.

Note in this option will also run all collections. See the ‘‘COLLECTION’’ section below.

−X chk1,chk2,..., **−−dont−check−part** chk1,chk2,...
Run all but the specified checks. You can either specify the name of the check script or the abbreviation. For details, see the ‘‘CHECKS’’ section below.

General options:

−h, **−−help**
Display usage information and exit.

−q, **−−quiet**
Suppress all informational messages including override comments (normally shown with **−−show−overrides**).

This option is silently ignored if **−−debug** is given. Otherwise, if both **−−verbose** and **−−quiet** is used, the last of these two options take effect.

This option overrides the **verbose** and the **quiet** variable in the configuration file. In the configuration file, this option is enabled by using **quiet** variable. The **verbose** and **quiet** variables may not both appear in the config file.

−v, **−−verbose**
Display verbose messages.

If **−−debug** is used this option is always enabled. Otherwise, if both **−−verbose** and **−−quiet** is used (and **−−debug** is not used), the last of these two options take effect.

This option overrides the **quiet** variable in the configuration file. In the configuration file, this option is enabled by using **verbose** variable. The **verbose** and **quiet** variables may not both appear in the config file.

−V, **−−version**
Display lintian version number and exit.

−−print−version
Print unadorned version number and exit.

Behavior options for **lintian**.

−−color (never|always|auto|html)
Whether to colorize tags in lintian output based on their severity. The default is "never", which never uses color. "always" will always use color, "auto" will use color only if the output is going to a terminal, and "html" will use HTML <span> tags with a color style attribute (instead of ANSI color escape sequences).

This option overrides the **color** variable in the configuration file.

−−default−display−level
Reset the current display level to the default. Basically, this option behaves exactly like passing the following options to lintian:

```
B<-L> ">=important" B<-L> "+>=normal/possible" B<-L> "+minor/certain"
```

The primary use for this is to ensure that lintian's display level has been reset to the built-in default values. Notably, this can be used to override display settings earlier on the command-line or in the lintian configuration file.

Further changes to the display level can be done *after* this option. Example: **−−default−display−level −−display−info** gives you the default display level plus informational ("I:") tags.

−−display−source X
Only display tags from the source X (e.g. the Policy Manual or the Developer Reference). This option can be used multiple times to add additional sources. Example sources are "policy" or "devref" being the Policy Manual and the Developer Reference (respectively).

The entire list of sources can be found in *$LINTIAN_ROOT/data/output/manual−references*

−E, **−−display−experimental**, **−−no−display−experimental**
Control whether to display experimental ("X:") tags. They are normally suppressed.

If a tag is marked experimental, this means that the code that generates this message is not as well tested as the rest of Lintian, and might still give surprising results. Feel free to ignore Experimental messages that do not seem to make sense, though of course bug reports are always welcome (particularly if they include fixes).

These options overrides the **display-experimental** variable in the configuration file.

−i, **−−info**
Print explanatory information about each problem discovered in addition to the lintian error tags. To print a long tag description without running lintian, see **lintian−info** (1).

This option overrides **info** variable in the configuration file.

−I, **−−display−info**
Display informational ("I:") tags as well. They are normally suppressed. (This is equivalent to **−L** ">=wishlist").

This option overrides the **display-info** variable in the configuration file.

Note: **display-level** and **display-info** may not both appear in the configuration file.

−L [+|−|=][>=|>|=|<|<=][S|C|S/C], **−−display−level** [+|−|=][>=|>|=|<|<=][S|C|S/C]
Fine-grained selection of tags to be displayed. It is possible to add, remove or set the levels to display, specifying a severity (S: serious, important, normal, minor, wishlist, pedantic), a certainty (C: certain, possible, wild-guess), or both (S/C). The default settings are equivalent to **−L** ">=important" **−L** "+>=normal/possible" **−L** "+minor/certain").

The value consists of 3 parts, where two of them are optional. The parts are:

modifier operator
How to affect the current display level. Can be one of add to ("+"), remove from ("−") or set to ("=") the display level(s) denoted by the following selection.

The default value is "=" (i.e. set the display level).

set operator
The set of severity and certainties to be selected. The operator can be one of ">=", ">", "=", "<" or "<=". As an example, this can be used to select all important (or "more severe") tags via ">=important".

The selected values includes only items where *both* the severity and the certainty are both included in the set.

As an example, ">=important/possible" includes "important/possible", "important/certain", "serious/possible" and "serious/certain". Note that it does *not* include "serious/wild−guess" (since it does not satisfy the lower-bound for the certainty).

The default value is "=", which means "exactly" the given severity or/and certainty.

severity-certainty
The severity or/and certainty. This can be any of the 3 forms: *severity*, *certainty* or *severity/certainty*.

If only a severity or a certainty is given, the other one defaults to "any" as in "any certainty with the given severity" (or vice versa).

This option overrides the **display-level** variable in the configuration file. The value of the **display-level** in configuration file should be space separated entries in the same format as passed via command-line.

Note: **display-level** may not be used with **display-info** or **pedantic** in the configuration file.

−o, **−−no−override**
Ignore all overrides provided by the package. This option will overrule **−−show−overrides**.

This option overrides the **override** variable in the configuration file.

−−pedantic
Display pedantic ("P:") tags as well. They are normally suppressed. (This is equivalent to **−L** "+=pedantic").

Pedantic tags are Lintian at its most pickiest and include checks for particular Debian packaging styles and checks that many people disagree with. Expect false positives and Lintian tags that you don't consider useful if you use this option. Adding overrides for pedantic tags is probably not worth the effort.

This option overrides the **pedantic** variable in the configuration file.

Note: **pedantic** and **display-level** may not both appear in the configuration file.

−−profile vendor[/prof]

Use the profile from vendor (or the profile with that name). If the profile name does not contain a slash, the default profile for than vendor is chosen.

As an example, if you are on Ubuntu and want to use Lintian's Debian checks, you can use:

```
  --profile debian
```

Likewise, on a Debian machine you can use this to request the Ubuntu checks.

If the token *{VENDOR}* appears in the profile name, **lintian** will substitute the token with a vendor name to find the profile. **lintian** uses Dpkg::Vendor to determine the best vendor to use (the closer to the current vendor, the better). This is mostly useful for people implementing their own checks on top of Lintian.

If not specified, the default value is *{VENDOR}/main*.

Please Refer to the Lintian User Manual for the full documentation of profiles.

−−show−overrides, **−−hide−overrides**

Controls whether tags that have been overridden should be shown.

The **−−show−overrides** differs from **−−no−overrides** in that shown overridden tags will still be marked as overridden (using an "O" code).

If the overridden tags are shown, the related override comments will also be displayed (unless −−quiet is used). Please refer to the Lintian User Manual for the documentation on how lintian relates comments to a given override.

These options override the **show-overrides** variable in the configuration file.

−−suppress−tags tag1,tag2,...

Suppress the listed tags. They will not be reported if they occur and will not affect the exit status of Lintian. This option can be given multiple times and can be mixed with **−−suppress−tags−from−file**.

This option can be used together with **−−dont−check−part** ("Not those checks nor these tags") and **−−check−part** ("Only those checks, but not these tags (from those checks)") to further reduce the selection of tags.

When used with **−−tags**, this option is mostly ignored.

−−suppress−tags−from−file file

Suppress all tags listed in the given file. Blank lines and lines beginning with # are ignored. All other lines are taken to be tag names or comma-separated lists of tag names to suppress. The suppressed tags will not be reported if they occur and will not affect the exit status of Lintian.

Tags parsed from the file will be handled as if they had been given to the **−−suppress−tags** option (e.g. ignored if **−−tags** is used).

−−tag−display−limit[=X]

By default, lintian limits itself to emitting at most 4 instances of each tag per processable when STDOUT is a TTY. This option specified that limit. See also **−−no−tag−display−limit**.

−−no−tag−display−limit

By default, lintian limits itself to emitting at most 4 instances of each tag per processable when STDOUT is a TTY. This option disables that limit.

When STDOUT is not a TTY, lintian has no limit. See also **−−tag−display−limit**.

Configuration options:

−−cfg configfile
Read the configuration from configfile rather than the default locations. This option overrides the **LINTIAN_CFG** environment variable.

−−no−cfg
Do not read any configuration file. This option overrides the **−−cfg** above.

−−ignore−lintian−env
Ignore all environment variables starting with *LINTIAN_*.

This option is mostly useful for applications running **lintian** for checking packages and do not want the invoking user to affect the result (by setting LINTIAN_PROFILE etc.).

Note it does *not* cause **lintian** to ignore the entire environment like *TMPDIR* or *DEB_VENDOR*. The latter can affect the default profile (or "{VENDOR}" token for **−−profile**).

Should usually be combined with **−−no−user−dirs** (or unsetting `$HOME` and all *XDG_* variables).

−−include−dir dir
Use dir as an additional "LINTIAN_ROOT". The directory is expected have a similar layout to the LINTIAN_ROOT (if it exists), but does not need to be a full self-contained root.

lintian will check this directory for (additional) profiles, data files, support libraries and checks. The latter two imply that Lintian may attempt to *load and execute code* from this directory.

This option may appear more than once; each time adding an additional directory. Directories are searched in the order they appear on the command line.

The additional directories will be checked *after* the user directories (though see **−−no−user−dirs**) and *before* the core LINTIAN_ROOT.

Note: This option should be the very first if given.

−j X, **−−jobs**=X
Set the limit for how many unpacking jobs Lintian will run in parallel. This option overrides the **jobs** variable in the configuration file.

By default Lintian will use *nproc* to determine a reasonable default (or 2, if the nproc fails).

−−user−dirs, **−−no−user−dirs**
By default, **lintian** will check *`$HOME`* and */etc* for files supplied by the user or the local sysadmin (e.g. config files and profiles). This default can be disabled (and re-enabled) by using **−−no−user−dirs** (and **−−user−dirs**, respectively).

These options will *not* affect the inclusion of LINTIAN_ROOT, which is always included.

These option can appear multiple times, in which case the last of them to appear determines the result.

Note that if the intention is only to disable the user's *`$HOME`*, then unsetting *`$HOME`* and *XDG_*_HOME* may suffice. Alternatively, */etc* can be "re-added" by using *−−include−dir* (caveat: */etc/lintianrc* will be ignored by this).

If the intention is to avoid (unintentional) side-effects from the calling user, then this option could be combined with **−−ignore−lintian−env**.

If for some reason **−−no−user−dirs** cannot be used, then consider unsetting *`$HOME`* and all the *`$XDG_`** variables (not just the *`$XDG_`*_HOME* ones).

Note: This option should be the very first if given.

Developer/Special usage options:

−−allow−root
Override lintian’s warning when it is run with superuser privileges.

−−keep−lab
By default, temporary labs will be removed after lintian is finished. Specifying this options will leave the lab behind, which might be useful for debugging purposes. You can find out where the temporary lab is located by running lintian with the **−−verbose** option.

−−packages−from−file X
The line is read as the path to a file to process (all whitespace is included!).

If X is “−”, Lintian will read the packages from STDIN.

−−perf−debug
Enable performance related debug logging.

The data logged and the format used is subject to change with every release.

Note that some of the information may also be available (possibly in a different format) with the **−−debug** option.

−−perf−output OUTPUT
Write performance related debug information to the specified file or file descriptor. If OUTPUT starts with a ’&’ or ’+’, Lintian will handle OUTPUT specially. Otherwise, Lintian will open the file denoted by OUTPUT for writing (truncating if it exists, creating it if it does not exist).

If the first character of OUTPUT is a & and the rest of argument is a number N, then lintian attempts to write it to the file descriptor with the number N. Said file descriptor must be open for writing. E.g *&2* makes Lintian write the performance logging to STDERR.

If the first character of OUTPUT is a +, Lintian will append to the file rather than truncating it. In this case, the file name is OUTPUT with initial “+” character removed. E.g. *+my−file* makes Lintian append to *my-file*

If Lintian should write the output to a file starting with a literal ’&’ or ’+’, then simply prefix it with “./” (e.g. “+my−file” becomes “./+my−file”).

If this option omitted, Lintian will default to using STDOUT.

−U info1,info2,..., **−−unpack−info** info1,info2,...
Collect information info1, info2, etc. even if these are not required by the checks. Collections requested by this option are also not auto-removed (in this run).

This option is mostly useful for debugging or special purpose setups.

It is allowed to give this option more than once. The following two lines of arguments are semantically equivalent:

```
 −U info1 −U info2
 −U info1,info2
```

CHECKS

apache2
Checks various build mistakes in Apache2 reverse dependencies

application-not-library
application packaged like a library (imported from pkg-perl-tools)

appstream-metadata
This script checks the AppStream metadata files for problems.

automake (**autom**)
Checks for erroneous, missing or deprecated automake files

binaries (**bin**)
This script checks binaries and object files for bugs.

changes-file (**chng**)
This script checks for various problems with .changes files

conffiles (**cnf**)
This script checks if the conffiles control file of a binary package is correct.

control-files (**ctl**)
Checks files in the binary package's control.tar.gz, etc.

cruft (**deb**)
This looks for cruft in Debian packaging or upstream source

dbus
Checks for deprecated or harmful D−Bus configuration

deb-format (**dfmt**)
This script checks the format of the deb ar archive itself.

debhelper (**dh**)
This looks for common mistakes in debhelper source packages.

duplicate-files (**dupf**)
This script checks for duplicate files using checksums

elpa (**elpa**)
This script checks if the packages comply with various aspects of the emacsen team's policy.

fields (**fld**)
This script checks the syntax of the fields in package control files, as described in the **Policy Manual**.

filename-length (**flen**)
This script checks for long package file names

files (**fil**)
This script checks if a binary package conforms to policy WRT to files and directories.

gir Checks for GObject-Introspection mini-policy compliance

group-checks (**gchck**)
This script checks for some issues that may appear in packages built from the same source. This includes intra-source circular dependencies and intra-source priority checks.

huge-usr-share (**hus**)
This script checks whether an architecture-dependent package has large amounts of data in */usr/share*.

infofiles (**info**)
This script checks if a binary package conforms to info document policy.

init.d (**ini**)
Check if a binary package conforms to policy with respect to scripts in */etc/init.d*.

java (**java**)
This script checks if the packages comply with various aspects of the Debian Java policy.

manpages (**man**)
This script checks if a binary package conforms to manual page policy.

md5sums (**md5**)
This script checks if md5sum control files are valid if they are provided by a binary package.

menu-format (**mnf**)
This script validates the format of **menu** files.

menus (**men**)
Check if a binary package conforms to policy with respect to **menu** and **doc-base** files.

nmu (**nmu**)
This script checks if a source package is consistent about its NMU-ness.

nodejs
This script checks nodejs-related issues

obsolete-sites (**obso**)
This script checks for obsolete (but still valid) URLs

ocaml (**ocaml**)
This looks for common mistakes in OCaml binary packages.

patch-systems (**pat**)
This script checks for various possible problems when using patch systems

pe This script checks Microsoft Windows Portable Executable (PE) files

phppear (**phppear**)
This script checks if the packages comply with various aspects of the Debian PHP policy.

python
This script checks Python-related issues

scripts (**scr**)
This script checks the #! lines of scripts in a package.

shared-libs (**shl**)
This script checks if a binary package conforms to shared library policy.

symlinks (**sym**)
This script checks for broken symlinks.

systemd
Checks various systemd policy things

testsuite
This script checks the Testsuite field in package dsc files, and *debian/tests/control* if any.

triggers
Check of trigger files in the binary package.

udev
This script checks the udev rules for problems.

upstream-metadata
This script checks the *upstream/metadata* file for problems.

upstream-signing-key
This script looks for public upstream signing keys in source packages and, if present, checks that they conform to current standards.

Each source package for which upstream developers sign releases should have a watch file and a public key to verify the sources retrieved from downloads.

usrmerge (**usr**)
This script checks for files with the same name installed in */* and */usr*.

COLLECTION

ar-info
This script runs the ‘‘ar t’’ command over all .a files of package.

bin-pkg-control
This script extracts the contents of control.tar into the *control/* and creates control-index as well.

changelog-file
: This script copies the *changelog* file and *NEWS.Debian* file (if any) of a package into the lintian directory.

copyright-file
: This script copies the *copyright* file of a package into the lintian directory.

diffstat
: This script extracts the Debian diff of a source package, and runs diffstat on it, leaving the result in the diffstat output file

file-info
: This script runs the **file** (1) command over all files of any kind of package.

java-info
: This script extracts information from manifests of JAR files

md5sums
: This script runs the **md5sums** (1) over all files in a binary package.

objdump-info
: This script runs **objdump** (1) over all binaries and object files of a binary package.

override-file
: This script copies the *override* file of a package into the lintian directory.

scripts
: This script scans a binary package for scripts that start with #! and lists their filenames together with the interpreter named by their first line.

```
 The format is: scriptpath filename
```

Note that the filename might contain spaces, but the scriptpath will not, because linux only looks at the first word when executing a script.

src-orig-index
: This script create an index file of the contents of the orig tarballs.

strings
: This script runs the **strings** (1) command over all files of a binary package.

unpacked
: This script unpacks the package under the *unpacked/* directory

FILES

Lintian looks for its configuration file in the following locations:

- The argument given to **--cfg**
- *$LINTIAN_CFG*
- *$XDG_CONFIG_HOME/lintian/lintianrc*
- *$HOME/.lintianrc*

  Deprecated in Lintian/2.5.12 and newer (use the XDG based variant above)
- *XGD_DIR/lintian/lintianrc*

  Where XGD_DIR is a directories listed in *$XDG_CONFIG_DIRS* (or */etc/xdg* if *$XDG_CONFIG_DIRS* is unset).
- */etc/lintianrc*

  Deprecated in Lintian/2.5.12 and newer (use the XDG based variant above)

Lintian uses the following directories:

/tmp
: Lintian defaults to creating a temporary lab directory in */tmp*. To change the directory used, set the TMPDIR environment variable to a suitable directory. TMPDIR can be set in the configuration file.

/usr/share/lintian/checks
: Scripts that check aspects of a package.

/usr/share/lintian/collection
: Scripts that collect information about a package and store it for use by the check scripts.

/usr/share/lintian/data
: Supporting data used by Lintian checks and for output formatting.

/usr/share/lintian/lib
: Utility scripts used by the other lintian scripts.

For binary packages, Lintian looks for overrides in a file named *usr/share/lintian/overrides/<package>* inside the binary package, where *<package>* is the name of the binary package. For source packages, Lintian looks for overrides in *debian/source/lintian−overrides* and then in *debian/source.lintian−overrides* if the first file is not found. The first path is preferred. See the Lintian User's Manual for the syntax of overrides.

CONFIGURATION FILE

The configuration file can be used to specify default values for some options. The general format is:

```
option = value
```

All whitespace adjacent to the "=" sign as well as leading and trailing whitespace is ignored. However whitespace within the value is respected, as demonstrated by this example:

```
# Parsed as "opt1" with value "val1"
   opt1    =   val1
# Parsed as "opt2" with value "val2.1  val2.2     val2.3"
opt2 = val2.1  val2.2     val2.3
```

Unless otherwise specified, no option may appear more than once. Lintian will ignore empty lines or lines starting with the #−character.

Generally options will be the long form of the command-line option without the leading dashes. There some exceptions (such as −−profile), where Lintian uses the same name as the environment variable.

Lintian only allows a subset of the options specified in the configuration file; please refer to the individual options in "OPTIONS".

In the configuration file, all options listed must have a value, even if they do not accept a value on command line (e.g. −−pedantic). The values "yes", "y", "1", or "true" will enable such an option and "no", "n", "0" or "false" will disable it. Prior to the 2.5.2 release, these values were case sensitive.

For other options, they generally take the same values as they do on the command line. Though some options allow a slightly different format (e.g. −−display−level). These exceptions are explained for the relevant options in "OPTIONS".

Beyond command line options, it is also allowed to specify the environment variable "TMPDIR" in the configuration file.

A sample configuration file could look like:

```
# Sample configuration file for lintian
#
# Set the default profile (--profile)
LINTIAN_PROFILE = debian

# Set the default TMPDIR for lintian to /var/tmp/lintian
# - useful if /tmp is tmpfs with "limited" size.
TMPDIR = /var/tmp/lintian/
```

```
# Show info (I:) tags by default (--display-info)
#  NB: this cannot be used with display-level
display-info=yes

# Ignore all overrides (--no-override)
#  NB: called "override" in the config file
#      and has inverted value!
override = no

# Automatically determine if color should be used
color = auto
```

EXIT STATUS

0 No policy violations or major errors detected. (There may have been warnings, though.)

1 Policy violations or major errors detected.

2 Lintian run-time error. An error message is sent to stderr.

CHECKING LAST BUILD

When run in an unpacked package dir (with no package selection arguments), Lintian will use *debian/changelog* to determine the source and version of the package. Lintian will then attempt to find a matching *.changes* file for this source and version combination.

Lintian will (in order) search the following directories:

.. Used by **dpkg−buildpackage** (1).

../build−area
  Used by **svn−buildpackage** (1).

/var/cache/pbuilder/result
  Used by **pbuilder** (1) and **cowbuilder** (1).

In each directory, Lintian will attempt to find a *.changes* file using the following values as architecture (in order):

$DEB_BUILD_ARCH (or *dpkg −−print−architecture*)
  The environment variable DEB_BUILD_ARCH (if not set, "dpkg −−print−architecture" will be used instead)

$DEB_HOST_ARCH
  The environment variable DEB_HOST_ARCH.

dpkg −−print−foreign−architectures
  If **dpkg** (1) appears to support multi-arch, then any architecture listed by "dpkg −−print−foreign−architectures" will be used (in the order returned by dpkg).

multi
  Pseudo architecture used by **mergechanges** (1).

all Used when building architecture indep packages only (e.g. dpkg-buildpackage −A).

source
  Used for "source only" builds (e.g. dpkg-buildpackage −S).

If a *.changes* file matches any combination above exists, Lintian will process the first match as if you had passed it per command line. If no *.changes* file can be found, Lintian will print a list of attempted locations on STDERR and exit 0.

EXAMPLES

$ lintian foo.changes
  Check the changes file itself and any (binary, udeb or source) package listed in it.

$ lintian foo.deb
Check binary package foo given by foo.deb.

$ lintian foo.dsc
Check source package foo given by foo.dsc.

$ lintian foo.dsc −L +minor/possible
Check source package foo given by foo.dsc, including minor/possible tags.

$ lintian −i foo.changes
Check the changes file and, if listed, the source and binary package of the upload. The output will contain detailed information about the reported tags.

$ lintian
Assuming *debian/changelog* exists, look for a changes file for the source in the parent dir. Otherwise, print usage information and exit.

BUGS

Lintian does not have any locking mechanisms yet. (Running several Lintian processes on the same laboratory simultaneously is likely to fail or corrupt the laboratory.)

If you discover any other bugs in lintian, please contact the authors.

SEE ALSO

lintian−info (1), Lintian User Manual (/usr/share/doc/lintian/lintian.html/index.html)

Packaging tools: **debhelper** (7), **dh_make** (8), **dpkg−buildpackage** (1).

AUTHORS

Niels Thykier <niels@thykier.net>

Richard Braakman <dark@xs4all.nl>

Christian Schwarz <schwarz@monet.m.isar.de>

Please use the email address <lintian−maint@debian.org> for Lintian related comments.