

**NAME**

Lintian::Check — Lintian checks shared between multiple scripts

**SYNOPSIS**

```
use Lintian::Check qw(check_maintainer);

my ($maintainer, $field) = ('John Doe', 'uploader');
check_maintainer ($maintainer, $field);
```

**DESCRIPTION**

This module provides functions to do some Lintian checks that need to be done in multiple places. There are certain low-level checks, such as validating a maintainer name and e-mail address or checking spelling, which apply in multiple situations and should be done in multiple checks scripts or in checks scripts and the Lintian front-end.

The functions provided by this module issue tags directly, usually either taking the tag name to issue as an argument or dynamically constructing the tag name based on function parameters. The caller is responsible for ensuring that all tags are declared in the relevant \*.desc file with proper descriptions and other metadata. The possible tags issued by each function are described in the documentation for that function.

**FUNCTIONS**

check\_maintainer(MAINTAINER, FIELD)

Checks the maintainer name and address MAINTAINER for Policy compliance and other issues. FIELD is the context in which the maintainer name and address was seen and should be one of `maintainer` (the `Maintainer` field in a control file), `uploader` (the `Uploaders` field in a control file), or `changed-by` (the `Changed-By` field in a changes file).

The following tags may be issued by this function. The string %s in the tags below will be replaced with the value of FIELD.

%s-address-is-on-localhost

The e-mail address portion of MAINTAINER is at `localhost` or some other similar domain.

= item %s-address-is-root-user

The user (from email or username) of MAINTAINER is `root`.

%s-address-causes-mail-loops-or-bounces

The e-mail address portion of MAINTAINER or UPLOADER refers to the PTS e-mail addresses `package@packages.debian.org` or `package@packages.qa.debian.org`, or, alternatively refers to a mailing list which is known to bounce off-list mails sent by Debian role accounts.

%s-address-looks-weird

MAINTAINER may be syntactically correct, but it isn't conventionally formatted. Currently this tag is only issued for missing whitespace between the name and the address.

%s-address-malformed

MAINTAINER doesn't fit the basic syntax of a maintainer name and address as specified in Policy.

%s-address-missing

MAINTAINER does not contain an e-mail address in angle brackets (<>).

%s-name-missing

MAINTAINER does not contain a full name before the address, or the e-mail address was not in angle brackets.

wrong-debian-qa-address-set-as-maintainer

MAINTAINER appears to be the Debian QA Group, but the e-mail address portion is wrong for orphaned packages. This tag is only issued for a FIELD of `maintainer`.

wrong-debian-qa-group-name

MAINTAINER appears to be the Debian QA Group, but the name portion is not Debian QA Group. This tag is only issued for a FIELD of maintainer.

The last two tags are issued here rather than in a location more specific to checks of the Maintainer control field because they take advantage of the parsing done by the rest of the function.

spelling\_tag\_emitter(TAGNAME[, FILENAME])

Create and return a subroutine that is useful for emitting lintian tags for spelling mistakes. The returned CODE ref can be passed to “check\_spelling(TEXT,[ EXCEPTIONS,] CODEREF)” and will faithfully emit TAGNAME once for each unique spelling mistake.

The optional extra parameter FILENAME is used to denote the file name, when this is not given from the tagname.

check\_spelling(TEXT,[ EXCEPTIONS,] CODEREF)

Performs a spelling check of TEXT. Call CODEREF once for each unique misspelling with the following arguments:

The misspelled word/phrase

The correct word/phrase

If EXCEPTIONS is given, it will be used as a hash ref of exceptions. Any lowercase word appearing as a key of this hash ref will never be considered a spelling mistake (exception being if it is a part of a multiword misspelling).

Returns the number of spelling mistakes found in TEXT.

check\_spelling\_picky(TEXT, CODEREF)

Performs a spelling check of TEXT. Call CODEREF once for each unique misspelling with the following arguments:

The misspelled word/phrase

The correct word/phrase

This method performs some pickier corrections – such as checking for common capitalization mistakes – which would be not included in check\_spelling as they are not appropriate for some files, such as changelogs.

Returns the number of spelling mistakes found in TEXT.

## VARIABLES

\$known\_shells\_regex

Regular expression that matches names of any known shell.

## AUTHOR

Originally written by Russ Allbery <rra@debian.org> for Lintian. Based on code from checks scripts by Marc Brockschmidt and Richard Braakman.

## SEE ALSO

**lintian**(1)