

**NAME**

Lintian::Tag::Info – Lintian interface to tag metadata

**SYNOPSIS**

```
my $cs = Lintian::CheckScript->new ("${ENV{'LINTIAN_ROOT'}}/checks/",
                                     'files');

my $tag_info = $cs->get_tag ('some-tag');
print "Tag info is:\n";
print $tag_info->description('text', ' ');
print "\nTag info in HTML is:\n";
print $tag_info->description('html', ' ');
```

**DESCRIPTION**

This module provides an interface to tag metadata as gleaned from the \*.desc files describing the checks. It can be used to retrieve specific metadata elements or to format the tag description.

**CLASS METHODS**

`new(HASH, SCRIPT_NAME, SCRIPT_TYPE)`  
Creates a new Lintian::Tag::Info.

**INSTANCE METHODS**

`load(PATH)`  
Loads a tag description from PATH.

`add_script_type(STRING)`  
Set script type to STRING. Only used to format output.

`certainty()`  
Returns the certainty of the tag.

`code()`  
Returns the one-letter code for the tag. This will be a letter chosen from E, W, I, or P, based on the tag severity, certainty, and other attributes (such as whether experimental is set). This code will never be O or X; overrides and experimental tags are handled separately.

`description([FORMAT [, INDENT]])`  
Returns the formatted description (the Info field) for a tag. FORMAT must be either `text` or `html` and defaults to `text` if no format is specified. If `text`, returns wrapped paragraphs formatted in plain text with a right margin matching the `Text::Wrap` default, preserving as verbatim paragraphs that begin with whitespace. If `html`, return paragraphs formatted in HTML.  
  
If INDENT is specified, the string INDENT is prepended to each line of the formatted output.

`experimental()`  
Returns true if this tag is experimental, false otherwise.

`severity([$real])`  
Returns the severity of the tag; if `$real` is a truth value the real (original) severity is returned, otherwise the effective severity is returned.

See `set_severity()`

`set_severity($severity)`  
Modifies the effective severity of the tag.

`script()`  
Returns the check script corresponding to this tag.

`sources()`  
Returns, as a list, the keywords for the sources of this tag from the references header. This is only the top-level source, not any more-specific section or chapter.

**tag()**

Returns the tag name.

**DIAGNOSTICS**

The following exceptions may be thrown:

no tag specified

The `Lintian::Tag::Info::new` constructor was called without passing a tag as an argument.

unknown output format %s

An unknown output format was passed as the `FORMAT` argument of `description()`. `FORMAT` must be either `text` or `html`.

The following fatal internal errors may be reported:

can't open %s: %s

The specified file, which should be part of the standard Lintian data files, could not be opened. The file may be missing or have the wrong permissions.

missing Check-Script field in %s

The specified check description file has no Check-Script field in its header section. This probably indicates the file doesn't exist or has some significant formatting error.

missing Tag field in %s

The specified check description file has a tag section that has no Tag field.

**FILES**

LINTIAN\_ROOT/checks/\*.desc

The tag description files, from which tag metadata is read. All files matching this shell glob expression will be read looking for tag data.

LINTIAN\_ROOT/data/output/manual-references

Information about manual references. Each non-comment, non-empty line of this file contains four fields separated by `: : .`. The first field is the name of the manual, the second field is the section or empty for data about the whole manual, the third field is the title, and the fourth field is the URL. The URL is optional.

**ENVIRONMENT**

LINTIAN\_ROOT

This variable specifies Lintian's root directory. It defaults to `/usr/share/lintian` if not set. The **lintian** program normally takes care of setting it.

**AUTHOR**

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

**SEE ALSO**

**lintian**(1)