The `changelog` Package

Rebecca Turner*

2019/07/06 2.1.0

Abstract

Changelogs are important. Unfortunately, there are few facilities for typesetting changelogs in LaTeX. `changelog` defines a `changelog` environment to make changelogs simple and intuitive.

For rationale, read keepachangelog.com.

NOTE Browse the sources, contribute, or complain at github.com/9999years/latex-changelog

Contents

---

*Brandeis University; rebeccaturner@brandeis.edu

1 I don't want to read this documentation

Perfectly understandable. `changelog` comes in a "simple" flavor:

Changelog

1.0.0 Rebecca Turner (2018-12-28)

- Cool features
- Bug fixes

0.1.0 Rebecca Turner (2018-10-25) — Initial beta

```
\begin{changelog}[author=Rebecca
    Turner, simple,
    sectioncmd=\section*]
\begin{version}[v=1.0.0,
    date=2018-12-28]
  \item Cool features
  \item Bug fixes
\end{version}
\shortversion{v=0.1.0,
    date=2018-10-25,
    changes=Initial beta}
\end{changelog}
```

And a more organized variant based on keepachangelog.com:

Changelog

1.0.0 Rebecca Turner (2018-12-28)

Added

- Cool features

Fixed

- Bug fixes

0.1.0 Rebecca Turner (2018-10-25) — Initial beta

```
\begin{changelog}[author=Rebecca
    Turner, sectioncmd=\section*]
\begin{version}[v=1.0.0,
    date=2018-12-28]
  \added
    \item Cool features
  \fixed
    \item Bug fixes
\end{version}
\shortversion{v=0.1.0,
    date=2018-10-25,
    changes=Initial beta}
\end{changelog}
```

2 Intro

2.1 Why?

Read Olivier Lacan's lovely site keepachangelog.com. To excerpt:

2.1.1 What is a changelog?

A changelog is a file which contains a curated, chronologically ordered list of notable changes for each version of a project.

2.1.2 Why keep a changelog?

To make it easier for users and contributors to see precisely what notable changes have been made between each release (or version) of the project.

2.1.3 Who needs a changelog?

People do. Whether consumers or developers, the end users of software are human beings who care about what's in the software. When the software changes, people want to know why and how.

2.2 The competition

It's always good to know the competition. Unfortunately, there isn't much here. Know of another package with similar functionality? Drop me a line or open a pull request!

vhistory provides a decent-looking changelog. However, it's designed for short changes and provides a less-than-elegant interface. Additionally, it's based on the `ltxtable` package, meaning it makes restrictions on the contents of version information and writes the table to a file.

holtxdoc has a decent changelog feature (via Oberdiek's `History` and `Version` environments), but `holtxdoc` "contains some private macros and setup for my needs. Thus do not use it." In addition, Oberdiek's changelogs don't support multiple authors.

gitlog is an interesting idea, but `gitlog` "is a proof-of-concept release to allow users an early evaluation…"

Also, friends don't let friends dump git logs into changelogs.

3 The changelog environment

`\begin{changelog}[⟨options⟩]...\end{changelog}`

Wraps `description` while providing the `version` environment and the `\shortversion` command. In addition to the options shown below, ⟨options⟩ may contain any of the options for `version` as a form of "partial application"; this may be useful if, for example, most of your versions have the same author.

section=⟨true|false⟩   true
Insert a `\section` before the changelog?

sectioncmd=⟨command⟩   \section
Which sectioning command to use?

title=⟨txt⟩   Changelog
What to title the changelog section?

label=⟨label⟩   sec:changelog
What to `\label` the section?

`\begin{version}[⟨options⟩]...\end{version}`

Gives a single version; wraps `itemize`.
If the `date` option is absent, the date isn't printed.
If the `v`/`version` option is absent, the date is used in its place.

If both `version` and `date` are absent, the version is shown as `Unreleased` and `\today` is used for the date. `\today` isn't ideal (which is to say, not ISO 8601 compliant) but it's well-known and easy to redefine.[1]

⟨*options*⟩ may include:

version=⟨*version string*⟩

    The version string for this version

v=⟨*version string*⟩

    An alias for `version`

author=⟨*author(s)*⟩

    The author(s) of this version

date=⟨*date string*⟩

    The date of this version's release

yanked=⟨true|false⟩   false

    Indicates that the release was revoked due to a "serious bug or security issue"; prints a visible notice next to the version number

simple=⟨true|false⟩   false

    Indicates this version isn't split up into `\added`, `\changed`, etc. categories; if this option is given, a `version` environment acts like a plain `itemize`

short=⟨true|false⟩   false

    Renders a short version, like `\shortversion`'s `changes` option; useful if you need to put verbatim material in the changes which would otherwise cause an error due to being used in an argument.

`\shortversion{`⟨*options*⟩`}`

    A short, one-line version. In addition to `version`'s options, the following options are available for `\shortversion`:

changes=⟨*change text*⟩

    The changes to display for this version

3.1   Helper commands

The changelog package defines several "helper commands." These commands introduce a *set* of changes within the version. There must be at least one `\item` between one of these commands and the end of the `version` environment.

    Note that these commands are *only* available when not using the `simple` option!

`\added`

    Introduces a list of `\item`s that represent added features

---

[1]Try the `datetime2` package or any of the other solutions here.

`\changed`

> Introduces a list of \items that represent changed features

`\deprecated`

> Introduces a list of \items that represent deprecated features

`\removed`

> Introduces a list of \items that represent features which have been removed

`\fixed`

> Introduces a list of \items that represent bug fixes

`\security`

> Introduces a list of \items that represent security-fixes and closed security holes

3.2 Customization

The changelog environment wraps changelogdescription (which is defined by default to be just the description environment), and the version environment wraps changelogitemize. One could customize these in depth with enumitem, as in:

```
\usepackage{enumitem}
\renewenvironment{changelogitemize}
    {\begin{itemize}[label=---]}
    {\end{itemize}}
```

`\changelogyanked`

> Prints the "revoked release" notice: **YANKED** . See: yanked releases on keepachangelog.com.

4 Translations

Want to use changelog in a non-English document? Great! Thanks to a contribution by Holger Schieferdecker (see section 5 below), changelog comes with translations for English, German, and support for adding new translations. To use built-in translations, simply load babel with the desired language:

1.0.0 Holger Schieferdecker
Hinzugefügt
• Features!

```
% In the preamble:
% \usepackage[german]{babel}

\begin{changelog}[section=false,
    author=Holger Schieferdecker]
  \begin{version}[v=1.0.0]
    \added
      \item Features!
  \end{version}
\end{changelog}
```

4.1 Creating new translations

changelog uses the translations package to provide translations, and a complete translation looks like this:

```
\documentclass{article}
\usepackage[english]{babel}
\usepackage{translations}
\DeclareTranslation{English}{changelog-Added}{Added}
\DeclareTranslation{English}{changelog-Changed}{Changed}
\DeclareTranslation{English}{changelog-Deprecated}{Deprecated}
\DeclareTranslation{English}{changelog-Removed}{Removed}
\DeclareTranslation{English}{changelog-Fixed}{Fixed}
\DeclareTranslation{English}{changelog-Security}{Security}
\DeclareTranslation{English}{changelog-Miscellaneous}{Miscellaneous}
\DeclareTranslation{English}{changelog-Unreleased}{Unreleased}
\DeclareTranslation{English}{changelog-Yanked}{YANKED}
\begin{document}
% ...
\end{document}
```

Refer to the translations package documentation and the babel package documentation for more information.

If you create a translation, please send me an email and I'll incorporate the translation into changelog's next release!

5 Contributors

Rebecca Turner Original implementation.

Holger Schieferdecker Internationalization support and German translations.

6 Changelog

This is this package's actual changelog — not an example!

2.1.0 Holger Schieferdecker (2019-06-29)

Added

• Internationalization support.

- German translations.

2.0.0 Rebecca Turner (2019-04-15)

Added

- Better error handling; more informative messages for empty `version` or `changelog` environments.
- `short` option in versions.

Removed

- Removed `color` package option.

Fixed

- Added dependencies for `xparse` and `xkeyval`, which had erroneously been excluded from the package.

1.0.0 Rebecca Turner (2018-12-28)

Added

- The "simple" option for changelogs which aren't split up into sections of added, changed, removed, etc. features

Changed

- Instead of commands like \added introducing an item, they introduce a list of items; this is what keepachangelog.com actually intended; previous implementations were incorrect to display an "Added" (or whatever) marker next to each item.

0.2.1 Rebecca Turner (2018-10-26) — Documented `yanked` option

0.2.0 Rebecca Turner (2018-10-26) — First stable release

0.1.0 Rebecca Turner (2018-10-25) — Initial beta