The changelog Package

Rebecca Turner[*]

2018-10-29

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:

```
\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}
```

Changelog

1.0.0 Rebecca Turner (2018-12-28)

- Cool features
- Bug fixes

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

And a more organized variant based on `keepachangelog.com`:

```
\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}
```

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

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

`changelog` [⟨*options*⟩]
Wraps `description` while providing the `version` environment and the `\shortversion` command. In addition to the options shown in table 1, ⟨*options*⟩ may contain any of the options for `version` (see figure 2) as a form of "partial application"; this may be useful if, for example, most of your versions have the same author.

Table 1: Options for the `changelog` environment

`section`	Insert a `\section` before the changelog? Default: true
`sectioncmd`	Which sectioning command to use? Default: `\section`
`title`	What to title the changelog section? Default: `Changelog`
`label`	What to `\label` the section? Default: `sec:changelog`

`version` [⟨*options*⟩]
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]

Table 2: Options for the `version` environment

`version`	The version string for this version
`v`	An alias for `version`
`author`	The author(s) of this version
`date`	The date of this version's release
`yanked`	Indicates that the release was revoked due to a "serious bug or security issue"; prints a visible notice next to the version number
`simple`	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`

`\shortversion` {⟨*options*⟩}
A short, one-line version. In addition to the options specified in table 2, the following options are available for `\shortversion`:

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

`changes` The changes to display for this version

3.1 Helper commands

The changelog package defines several “helper commands,” which are colorized if the `color` package option has been given. See section 3.3 for more information. 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.

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

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

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

`\removed` Introduces a list of `\item`s that represent features which have been removed

`\fixed` Introduces a list of `\item`s that represent bug fixes

`\security` Introduces a list of `\item`s 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`.

3.3 Colors

Colored output is supported by xcolor, which defines several named colors (as seen in figure 1); these colors may be redefined as needed. While the xcolor documentation goes into great detail, you’ll likely do fine with just e.g. `\colorlet{ChangelogAdded}{magenta}`.

3.3.1 Package options

`color` Makes output more colorful; this is probably not that useful

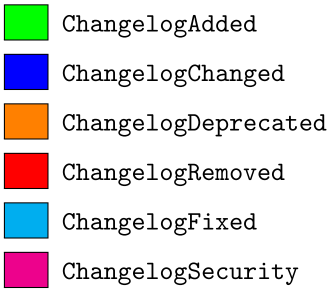

Figure 1: Colors provided by the `changelog` package

4 Changelog

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

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

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the definition; numbers in roman refer to the pages where the entry is used.

List of Tables