The screenwriter class*

J. A. Corbal†

Version 1.0.1 — 2026-05-09

Abstract

The screenwriter class provides a LaTeX document class for writing film and television screenplays. It is based on John Pate's `screenplay` class, updated for current LaTeX distributions and extended with modular language support and better integration with `babel`.

1 Introduction

The screenwriter class is a LaTeX document class designed to typeset film and television screenplays. It provides the conventional screenplay layout (12 pt monospaced type, specific margins, dialogue blocks, and sluglines) and a set of high-level commands tailored to the way screenwriters actually work.

The class is based on John Pate's original `screenplay.cls`, but has been updated and extended for current LaTeX systems. In particular, it modernises package loading, separates language-specific text into dedicated language definition files, and offers better cooperation with the `babel` package.

Unlike generic article or book classes, screenwriter focuses on the visual and structural conventions of screenplays: scene headings, dialogue layout, transitions such as "FADE IN:" and "FADE OUT:", and a standardised cover page. The goal is to let authors concentrate on the script itself, while LaTeX takes care of consistent formatting.

The class is intended both for individual writers who like working with LaTeX, and for production environments where reproducible layouts and multi-language support are important.

2 Installation

This section describes a typical local installation in a personal TeX tree. System-wide installation follows the same pattern, but requires administrator privileges.

The distribution of screenwriter consists of the following types of files:

*This document was last revised and validated on May 10, 2026.

†`jacorbal [at] gmail [dot] com`

- `screenwriter.cls`: the main document class.
- `screenwriter-lang-<`*`language`*`>.ldf`: language definition files containing all language-dependent strings used by the class (e.g., “FADE IN”, “THE END”, “INT.”, “EXT.”).
- `screenwriter-doc.tex` & `screenwriter-doc.pdf`: this documentation and its source.
- `LICENSE`: license file (LPPL 1.3c).
- `README`: this file.

A typical TEX directory structure might look like:

```
screenwriter/
  LICENSE
  README
  doc/
    screenwriter-doc.tex
    screenwriter-doc.pdf
  tex/
    latex/
      screenwriter/
        screenwriter.cls
        lang/
          screenwriter-lang-english.ldf
          screenwriter-lang-spanish.ldf
          ...
```

For a local installation, copy the `tex/latex/screenwriter/` contents into your personal TEX tree, for example:

```
$TEXMFHOME/tex/latex/screenwriter/
```

where `$TEXMFHOME` is usually `~/texmf` on a standard TEX Live installation. The language definition files can be placed in a `lang/` subdirectory as shown above; the class looks for them as `lang/screenwriter-lang-<`*`language`*`>.ldf`.

After copying the files, update your file name database if required by your distribution (for example by running `mktexlsr` on TEX Live). Once this is done, LATEX will be able to find the class and language files automatically, and you can use in any document:

```
\documentclass{screenwriter}
```

3 Basic usage

This section shows a minimal example and explains the main elements that screenwriter provides out of the box.

A minimal document

A very small English screenplay using the default layout might look as follows:

```
\documentclass{screenwriter}

\title{My First Screenplay}
\author{Screenwriter Name}

\begin{document}
    \coverpage

    \fadein
    \intslug[DAY]{OFFICE}

    \begin{dialogue}{WRITER}
        This is a line of dialogue.
    \end{dialogue}

    \theend
\end{document}
```

The class sets up 12 pt monospaced type, appropriate margins and dialogue widths, and a standard page style. The example uses only a few of the commands provided, but already shows the typical structure: cover page, opening transition, first scene heading, dialogue, and a final “THE END” marker.

Basic structure

In a typical screenplay document:

- The preamble selects the class and language, using `\documentclass[⟨options⟩]{screenwriter}`, and provides meta-data such as `\title` and `\author`.
- The document body starts with `\coverpage` or `\nicholl` to produce a title page.
- Each scene begins with a slugline, such as `\intslug` or `\extslug`.
- Dialogue is set within the `dialogue` environment, with optional parentheticals and continuation marks handled by `\dialbreak`.
- Transitions such as `\fadein`, `\fadeout`, `\intercut` and `\flashback` are used where needed.
- The script can end with `\theend`, which inserts the language-dependent “THE END” marker centered on the page.

4 Options

The screenwriter class accepts a small set of options that control page layout, screenplay style, and language.

4.1 Paper and layout options

The class internally loads the standard `article` class with 12 pt, one-column, one-sided settings, and then adjusts margins and text width to match common screenplay practice.

Two paper size options are available:

`letterpaper` Use US Letter paper (default). This is the usual choice for North American markets.

`a4paper` Use A4 paper and adjust the text and dialogue widths accordingly. This is often more convenient for European printers and readers.

The screenplay style can be selected with:

`literary` Literary (spec) screenplay style. Scene headings (sluglines) are printed without scene numbers. This is the default.

`technical` Technical (numbered) style. Each scene heading is automatically numbered, and the scene number is printed in the margins. This is useful in production and shooting scripts.

Examples:

```
\documentclass[a4paper]{screenwriter}
\documentclass[letterpaper,technical]{screenwriter}
```

4.2 Language options

The class separates language-dependent strings (such as “FADE IN”, “THE END”, “INT.”, “EXT.”, “TITLE OVER”, etc.) into external language definition files `screenwriter-lang-⟨language⟩.ldf`. The language used by the class can be selected in two ways.

Explicit language selection

You can explicitly choose the language via class options. The following languages are currently provided[1]:

`english`, `spanish`, `french`, `german`, `italian`, `portuguese`, `galician`, `catalan`, `esperanto`

For example:

[1] The language files should undergo a formal review to ensure accuracy and consistency.

```
\documentclass[spanish]{screenwriter}
```

tells the class to use Spanish strings and to therefore load the language file `lang/screenwriter-lang-spanish.ldf`, regardless of whether the document uses `babel` or not.

Cooperation with `babel`

If no explicit language option is given to screenwriter, but the user loads `babel` with a language, the class will attempt to detect the current `babel` language via `\languagename` at `\begin{document}`. It then tries to load the corresponding language file:

```
\documentclass{screenwriter}
\usepackage[french]{babel}
```

In this case, the class will look for `lang/screenwriter-lang-french.ldf` and use French strings for all its internal labels. If no matching language file is found, the class falls back to English.

Note. The class screenwriter does *not* load `babel` on its own; users are free to choose `babel`, `polyglossia`, or no multilingual package at all. The class only mirrors the active language when possible.

5 Commands

This section summarises the main commands and environments provided by the screenwriter class. They are grouped by function: title and address information, sluglines (scene headings), dialogue, and transitions.

5.1 Title and cover page

The class reuses the standard LaTeX meta-data commands `\title` and `\author`, and adds a few macros specific to screenplays.

`\title{<title>}` Sets the script title. This is used on the cover page and may also appear on the first page of the script.

`\author{<name>}` Sets the author name shown beneath the title.

`\realauthor{<name>}` Sets the real author name, as printed in the address block on the cover when `\author` refers to a pseudonym. By default, this is the same as `\author`.

`\address{<text>}` Sets the contact address that appears in the cover page address block. The default is a generic “Contact via Agency” (or its language-dependent equivalent).

`\agent{<text>}` Sets the agent information for the cover page. By default, this is left empty.

Once these values are set, you can produce a cover page using:

`\coverpage` Typesets a standard cover page: title in uppercase, "by" (or its translation) and author name, and an address block containing agent and author contact information. The page counter is reset to 1 afterwards.

`\nicholl` Produces a simpler title page variant with just the title centered on the page and no address block. The page counter is also reset to 1.

5.2 Sluglines (scene headings)

Sluglines mark the beginning of scenes, typically with "INT." or "EXT.", the location, and time of day. The class offers commands that build upon the language-dependent abbreviations defined in the language files.

`\intslug[⟨qualifier⟩]{⟨location⟩}` Interior slugline. The class prints the language-specific "INT" abbreviation, followed by the location and any qualifier (such as "DAY", "NIGHT"). The optional `⟨qualifier⟩` argument is placed at the end of the slugline, separated by a symbol, usually a dash.

`\extslug[⟨qualifier⟩]{⟨location⟩}` Exterior slugline. As above, but using the "EXT" abbreviation.

`\intextslug[⟨qualifier⟩]{⟨location⟩}` Combined interior/exterior slugline ("INT./EXT."), useful for scenes that move between inside and outside.

`\extintslug[⟨qualifier⟩]{⟨location⟩}` Combined exterior/interior slugline ("EXT./INT.").

In `literary` mode, sluglines are printed without scene numbers. In `technical` mode, each slugline increments an internal scene counter and prints the scene number in the margins.

A typical usage might be:

```
\intslug{OFFICE -- DAY}
\extslug[NIGHT]{STREET}
```

5.3 Dialogue

Dialogue is set using a dedicated environment which handles character names, optional parenthetical remarks, and the width and indentation of dialogue blocks.

`\begin{dialogue}[⟨paren⟩]{⟨CHARACTER⟩} ... \end{dialogue}` Starts a dialogue block for the given character name. The character name is printed in uppercase at the standard dialogue position. The optional `⟨paren⟩` argument allows you to add a parenthetical remark (such as "whispering"), which is typeset in parentheses above the dialogue text.

Example:

```
\begin{dialogue}[whispering]{WRITER}
    This is a whispered line.
\end{dialogue}
```

`\dialbreak[⟨paren⟩]{⟨CHARACTER⟩}` Ends the current `dialogue` environment with a "(MORE)" marker, starts a new page, and opens a new dialogue block for the same character, labelled with a continuation marker (typically "(CONT'D)" or its translation). The optional `⟨paren⟩` argument works as for `dialogue`.

This is useful when a character's speech continues across a page break in the final layout.

Parenthetical remarks can also be inserted manually using:

`\paren{⟨text⟩}` Typesets a parenthetical remark in a narrower column, typically centered under the character name and above the next line of dialogue.

5.4 Transitions and labels

The class provides a number of commands for common transitions and on-screen text labels. Their exact spelling and punctuation are language-dependent and defined in the language files.

`\fadein` Inserts the language-specific "FADE IN:" label followed by a vertical skip. This is typically used at the beginning of the script.

`\fadeout` Inserts the language-specific "FADE OUT:" label (or its equivalent), usually aligned to the right in the standard transition position.

`\intercut` Inserts an "INTERCUT WITH:" label, usually flush right, to indicate intercutting between locations.

`\flashback` Inserts a "FLASHBACK TO:" label (or equivalent), again in the standard transition position.

For on-screen textual titles, use the `titleover` environment:

`\begin{titleover}[⟨qualifier⟩] ... \end{titleover}` Starts a "TITLE OVER:" block. The optional `⟨qualifier⟩` argument allows additional specification (for example, a location or person). The body of the environment is typeset as the on-screen text.

Within this environment, `\titbreak` can be used to mark a break to the next page while indicating that the title continues.

Finally, to mark the end of the script:

`\theend` Typesets the language-specific "THE END" (or equivalent) centered on the page.

6 Language files

All language-dependent strings used by the screenwriter class are defined in separate language files named `screenwriter-lang-⟨language⟩.ldf`. This keeps the core class code independent of any particular language and makes it easy to add new languages or adjust existing ones.

Structure of a language file

A typical language file has the following structure:

```
\ProvidesFile{screenwriter-lang-spanish.ldf}[2026/05/09 v0.1 Spanish]

\renewcommand*{\myaddresstext}{Agencia de contacto}
\renewcommand*{\mymoretext}{(M\'AS)}
\renewcommand*{\mycontdtext}{(CONT.)}
\renewcommand*{\myoffscreentext}{(F.P.)}
\renewcommand*{\myvoiceovertext}{(\emph{OFF})}
\renewcommand*{\myfadeintext}{ABRE DE NEGRO}
\renewcommand*{\myfadeouttext}{FUNDE A NEGRO}
\renewcommand*{\myadlibtext}{\emph{ad~lib.}}
\renewcommand*{\mypunctcharA}{:}
\renewcommand*{\mypunctcharB}{.}
\renewcommand*{\myplacesep}{.~}
\renewcommand*{\mysepintext}{./}
\renewcommand*{\mybytext}{por}
\renewcommand*{\myinttext}{INT}
\renewcommand*{\myexttext}{EXT}
\renewcommand*{\mytitleovertext}{R\'OTULO}
\renewcommand*{\myintercuttext}{CORTE A}
\renewcommand*{\myflashbacktext}{ANALEPSIS A}
\renewcommand*{\mypovtext}{P.D.V.}
\renewcommand*{\myreverttext}{REGRESO DE \pov}
\renewcommand*{\mythirtytext}{FIN}
\renewcommand*{\myslugspace}{ -- }
```

The file starts with a `\ProvidesFile` declaration, followed by a series of `\renewcommand*` statements that assign the language-specific strings to the internal macros used by the class. No packages are loaded and no options are processed in these files; they only contain definitions.

The class expects the following macros to be defined (or redefined) by each language file:

- Cover and address strings:
  `\myaddresstext`, `\mybytext`.
- Dialogue markers:
  `\mymoretext`, `\mycontdtext`, `\myoffscreentext`, `\myvoiceovertext`, `\myadlibtext`.

- Transitions and labels:
  `\myfadeintext`, `\myfadeouttext`, `\mytitleovertext`, `\myintercuttext`, `\myflashbacktext`.
- Scene heading components:
  `\myinttext`, `\myexttext`, `\myplacesep`, `\mysepintext`, `\myslugspace`.
- Other:
  `\mypovtext`, `\myreverttext`, `\mythirtytext`.

Customizing individual strings

In some cases, users may wish to keep a given language but customise the wording of one or two internal strings (for example, preferring "SMASH CUT TO:" instead of "CUT TO:"). Since all language-dependent text is stored in macros, this can be done directly in the document preamble.

For example, to change the "FADE IN" label while still using the English language file:

```
\documentclass[english]{screenwriter}

% custom wording for the opening transition
\renewcommand*{\fadeintext}{SMASH CUT IN}

\begin{document}
    \fadein
    ...
\end{document}
```

Likewise, to adjust only the "THE END" marker in Spanish:

```
\documentclass[spanish]{screenwriter}

% use a different final marker
\renewcommand*{\thirtytext}{FIN DE LA OBRA}

\begin{document}
    ...
    \theend
\end{document}
```

When using `babel`, these customisations should be placed in the preamble *after* loading the class and any multilingual packages, so that they override the values set by the language definition file.

Adding a new language

To add support for a new language, you only need to:

1. Create a new file `screenwriter-lang-⟨language⟩.ldf` in the `lang/` subdirectory, based on one of the existing language files.

2. Provide appropriate translations for all `\my...` macros.

3. Optionally, define a class option in `screenwriter.cls` so a new language can be selected explicitly, e.g.

```
\DeclareOption{⟨language⟩}{%
  \def\screenwriter@lang{⟨language⟩}%
  \screenwriter@lang@explicittrue}
```

If no explicit class option is given, but the document uses `babel` and the current `\languagename` matches the file name, screenwriter will automatically pick up the new language file without further changes to the class.

7 License and credits

This work may be distributed and/or modified under the conditions of the *LaTeX Project Public License* (LPPL), either version 1.3c of this license or (at your option) any later version.

The latest version of the LPPL is available at:

https://www.latex-project.org/lppl.txt

Feedback, bug reports, and suggestions for new language files are very welcome.

Credits

The design of screenwriter is based on John Pate's original `screenplay` class, which provided an early solution for typesetting screenplays in LaTeX.

The present class updates and extends that work for current LaTeX distributions, introduces modular language support, and refines several aspects of the layout and implementation. Any remaining mistakes or omissions are the responsibility of the current maintainer.