tlc-article Getting Started Guide

Gary Allan Howard

October 4, 2026

Abstract

The tlc-article "Getting Started Guide" explains how to install and use tlc-article, customize its layout, use its public commands, and identify its package dependencies.

Contents

1 Installation . . . **3**
1.1 Distribution installation . . . 3
1.2 Local installation . . . 3

2 General Use Case . . . **4**
2.1 Document Layout . . . 4
2.2 Document Class tlc-article . . . 4
2.3 Title, Author & Abstract . . . 4
2.4 Table of Contents . . . 5
2.5 Header & Footer . . . 5

3 Customization . . . **6**
3.1 data/additional-layout.tex . . . 6
3.2 data/header-footer.tex . . . 6
3.3 data/version.csv . . . 7
3.4 data/logo.png . . . 8

4 Definitions & Commands . . . **8**
4.1 tlcBeginLandscape . . . 8
4.2 tlcEndLandScape . . . 8
4.3 tlcDarkblue . . . 8
4.4 tlcTitlePageAndTableOfContents . . . 8
4.5 newcolumn type: L, C & R . . . 9
4.6 data/additional-layout.tex . . . 9
4.7 data/header-footer.tex . . . 9
4.8 data/version.csv . . . 9
4.9 data/logo.png . . . 9
4.10 tlcDebug . . . 10

5 Required Packages . . . **11**

1 Installation

Install tlc-article with your TEX distribution's package manager whenever possible. You can instead copy the class into an individual document project when you need a local installation or want to test a development version.

1.1 Distribution installation

TEX Live users can install tlc-article with `tlmgr`:

```
tlmgr install tlc-article
```

MiKTEX users can install tlc-article with the MiKTEX Console. Your distribution will place the class in the appropriate TEX tree and maintain its filename database.

1.2 Local installation

Download the release archive or clone the repository, then copy `tlc-article.cls` next to your document's `.tex` file. LATEX searches the current directory before installed package trees.

```
cp tlc-article.cls /path/to/document/
```

Note The files in the optional `data/` directory customize the class. Copy the directory into your document project when you want to use or adapt those examples.

2 General Use Case

The goal of tlc-article is to simplify document layout. tlc-article orchestrates a logical arrangement for document header, footer, author, abstract, table of contents, and margins. The following sections outline the default implementation for each part tlc-article organizes.

Note This document was typeset using the instructions provided throughout this section.

2.1 Document Layout



Figure 1: Document Layout

2.2 Document Class tlc-article

tlc-article extends the article document class. tlc-article provides options directly to the article document class. As an example, the author can specify the font as follows:

```
\documentclass[12pt]{tlc-article}
```

2.3 Title, Author & Abstract

tlc-article has a macro tlcTitlePageAndTableOfContents that can be used to set the document title, document author, and document abstract, and establish the Table of Contents. The sample below reveals how to use tlcTitlePageAndTableOfContents.

```
\tlcTitlePageAndTableOfContents
   {Document Title}
   {Document Author}
   {Document Abstract}
```

2.4 Table of Contents

The table of contents begins on the page after the title and abstract. Its links are dark blue, and dotted leaders separate section titles from page numbers.

2.5 Header & Footer

fancyhdr is used to render the header and the footer. The author can override tlc-article by providing an implementation in data/header-footer.tex or augment the tlc-article application by providing data/version.csv. The sections below show the placement tlc-article uses when writing objects, and where the objects are defined.

Note tlc-article ignores data/version.csv when data/header-footer.tex is defined.

Header

Left When data/logo.png is found, the logo.

Center The document title.

Right When data/version.csv is present, the status, date, and version.

Footer

Left When data/version.csv is present, the institution.

Center When data/version.csv is present, the permission or license.

Right The current and total page count.

Rule width

A 0.1pt rule width is placed below the document header and above the document footer.

3 Customization

This section describes how tlc-article can be customized by using the file-hooks tlc-article checks for. The tlc-article default implementation will be used when the file-hooks are not found.

Note tlc-article consumes data/additional-layout.tex and data/header-footer.tex while processing the preamble.

3.1 data/additional-layout.tex

tlc-article will use whatever LaTeX definitions are found in data/additional-layout.tex when it exists. The file-check is shown below:

```
\IfFileExists{data/additional-layout.tex}
  {\input{data/additional-layout.tex}}
  {}
```

3.2 data/header-footer.tex

In the absence of data/header-footer.tex, tlc-article has a built-in header and footer strategy that is based on *fancyhdr*, *titling*, and *lastpage* LaTeX packages. The default implementation is shown below:

```

\IfFileExists{\tlc@headerFooter}%
{% use the custom header and footer defined by \tlc@headerfooter
  \input{\tlc@headerFooter}%
}%
{% Else: use default header and footer
  \useDefaultHeaderFooter%
}%

\newcommand{\useDefaultHeaderFooter}{%
  \useLogoFile%              Typeset Logo in the left side of the header.
  \useTitle%                 Typeset the title in the center of the header.
  \useVersionFile%           Typeset version info in right side of the header.
  \useRuler%                 Typeset header and footer with a ruler.
}%

\newcommand{\useLogoFile}{%
  \IfFileExists{\tlc@logoFile}%
  {%
    \fancyhead[L]{\includegraphics[width=3cm,height=1cm]{\tlc@logoFile}}%
  }%
  {%
   % Else: no operation because tlc@logoFile does not exist.
  }%
}%

\newcommand{\useVersionFile}{%
  \IfFileExists{\tlc@versionFile}%
  {%
   % document status, document date and document version.
    \fancyhead[R]{\tiny \tlc@status \\ \tlc@date \\ \tlc@version}%
   %
   % document owner.  This may be a person or company name.
    \fancyfoot[L]{\tiny \tlc@institution}%
   %
   % document license. This may be a license or a word like confidential.
    \fancyfoot[C]{\tiny \tlc@permission}%
  }%
  {%
   % Else: no operation because tlc@versionFile does not exist.
  }%
}%

\newcommand{\useRuler}{%
  \renewcommand{\headrulewidth}{0.1pt}%
  \setlength\headheight{34.0pt}%
  \fancyfoot[R]%
    {\tiny%
      {Page \thepage~of~\pageref{LastPage}}%
    }%
  \renewcommand{\footrulewidth}{0.1pt}%
}%

\newcommand{\useTitle}{%
  \fancyhead[C]{\large{\thetitle}}%
}%
```

The default implementation can be overridden by defining data/header-footer.tex.
Note When data/header-footer.tex exists and is empty, the document uses the header and footer defaults inherited from the article class.

3.3 data/version.csv

tlc-article will populate the built-in header and footer with information extracted from data/version.csv when it is present. data/version.csv is a tabular text file that uses the pipe character as its field delimiter and has the following column names:

version
: The version value is typeset in the right header. This field is used to convey the document version when it reached its current state.

date
: The date value is typeset in the right header. This field is used to communicate when the document transitioned into its current state.

status
: The status value is typeset in the right header. This field is used to convey the document state such as Approved, Draft, Effective, or Obsolete.

institution
: The institution value is typeset in the left footer. This field is used to tell the reader the author name or company name.

permission
: The permission value is typeset in the center footer. This field is used to identify confidentiality or a particular license.

The extraction methods are shown below.

```
% Extract document status, document date and document version from
% \tlc@versionFile.
% Argument:
%   1 - the column name to extract from the data file.
\newcommand{\tlcVersionPart}[1]{
  \csvreader[separator=pipe]
  {\tlc@versionFile}{
    1=\version,
    2=\date,
    3=\status,
    4=\institution,
    5=\permission
  }{#1}
}%

% Define extractions macros when \tlc@versionFile exists.
\IfFileExists{\tlc@versionFile}
{
  \def\tlc@version{\tlcVersionPart{\version}}
  \def\tlc@date{\tlcVersionPart{\date}}
  \def\tlc@status{\tlcVersionPart{\status}}
  \def\tlc@institution{\tlcVersionPart{\institution}}
  \def\tlc@permission{\tlcVersionPart{\permission}}
}
{% Else: no operation because tlc@versionFile does not exist.
}
```

3.4 data/logo.png

tlc-article will place data/logo.png in the left header when it is present. Make sure your logo's height is not larger than 34pt to avoid the "Package Fancyhdr Warning: headheight is too small" warning.

tlc-article file-check is shown below:

```
% Typeset the logo in the left side of the document header.  Otherwise no
% operation because tlc@logoFile does not exist.

\newcommand{\useLogoFile}{%
  \IfFileExists{\tlc@logoFile}%
  {%
    \fancyhead[L]{\includegraphics[width=3cm,height=1cm]{\tlc@logoFile}}%
  }%
  {%
   % Else: no operation because tlc@logoFile does not exist.
  }%
}%
```

4 Definitions & Commands

4.1 tlcBeginLandscape

Page layout is rotated 90° clockwise resulting in a landscape page orientation. Landscape orientation remains active until tlcEndLandScape.

4.2 tlcEndLandScape

Page layout is returned to portrait orientation when tlcEndLandScape is reached.

4.3 tlcDarkblue

tlcDarkblue is used throughout this document to render text using rgb{0,0,0.5}. tlcDarkblue is safe to use within your document.

4.4 tlcTitlePageAndTableOfContents

tlcTitlePageAndTableOfContents creates the document layout shown in Figure 1. Section 2.3 shows an example implementation.

tlcTitlePageAndTableOfContents command is shown below:

```
% Each document uses the same author name, title page and table of contents.
%
% Arguments:
%   1 - the title
%   2 - the author
%   3 - the abstract

\newcommand{\tlcTitlePageAndTableOfContents}[3]{%
  \ifthenelse{\equal{#1}{}}{% title check
    \title{}%
  }{% Else: title
    \title{#1}%
  }%
  \ifthenelse{\equal{#2}{}}{% author check
    \author{}%
  }{% Else: author
    \author{#2}%
  }%
  \maketitle%
  \ifthenelse{\equal{#3}{}}{% abstract check
  }{% Else: abstract
    \begin{abstract}#3\end{abstract}%
  }%
  \thispagestyle{empty}%
  \clearpage%
  \tableofcontents%
```

```
  \clearpage%
}%
```

4.5 newcolumn type: L, C & R

The new newcolumn type: L, C & R column types are Left, Center, and Right, respectively, and are designed for use with longtable. Data is wrapped within a table cell. The parameter defines the column width. As an example, L{3.14159cm} yields a left aligned, ragged right, wrapped text within a 3.14159cm wide cell.

```
\newcolumntype{L}[1]{>{\raggedright\let\newline\\\arraybackslash}p{#1}}
\newcolumntype{C}[1]{>{\centering\let\newline\\\arraybackslash}p{#1}}
\newcolumntype{R}[1]{>{\raggedleft\let\newline\\\arraybackslash}p{#1}}
```

4.6 data/additional-layout.tex

data/additional-layout.tex is an architectural hook the author can use to provide packages and commands not provided by tlc-article and to design implementations that are specific to your document.

Note `https://github.com/Traap/autodoc` is a documentation framework that extends tlc-article.

4.7 data/header-footer.tex

data/header-footer.tex is an architectural hook the author should use to completely override header and footer layouts provided by tlc-article.

4.8 data/version.csv

data/version.csv is used by tlc-article to populate the document header & footer. Refer to section 3.3 for data/version.csv definitions. data/version.csv is not used by tlc-article when data/header-footer.tex is defined. However, you might want to use the version hook by defining data/version.csv and using the commands below to extract data from data/version.csv in your data/header-footer.tex.

1. tlc@version
2. tlc@date
3. tlc@status
4. tlc@institution
5. tlc@permission

4.9 data/logo.png

tlc-article places data/logo.png in your header when defined.

4.10 tlcDebug

The tlcDebug command can assist you when you encounter compilation errors using tlc-article.

```
% We define tlcDebug to aid our users when they are debugging their document.
% tlcDebug should be placed at the end of your document to allow LaTeX to
% fully expand all macros and definitions.

\newcommand{\tlcDebug}{%
  \clearpage%
  \section{tlc-article Debug}%
  \subsection{tlc-article default files}%
  \begin{description}[align=right,labelindent=5cm]%
    \item[tlc@location:]\tlc@location%
    \item[tlc@additionalLayout:]\tlc@additionalLayout%
    \item[tlc@headerFooter:]\tlc@headerFooter%
    \item[tlc@logoFile:]\tlc@logoFile%
    \item[tlc@versionFile:]\tlc@versionFile%
  \end{description}%
  %
  \subsection{tlc-article file hooks}%
  \begin{description}[align=right,labelindent=5cm]%
    \item [tlc@additionalLayout:]\tlcIsDefined{\tlc@additionalLayout}%
    \item [tlc@headerFooter:]\tlcIsDefined{\tlc@headerFooter}%
    \item [tlc@logoFile:]\tlcIsDefined{\tlc@logoFile}%
    \item [tlc@versionFile:]\tlcIsDefined{\tlc@versionFile}%
  \end{description}%
  %
  \subsection{tlc-article header and footer hooks}%
  \begin{description}[align=right,labelindent=5cm]%
    \item[tlc@version:] \tlc@version%
    \item[tlc@date:] \tlc@date%
    \item[tlc@status:] \tlc@status%
    \item[tlc@institution:] \tlc@institution%
    \item[tlc@permission:] \tlc@permission%
  \end{description}%
}%
```

5 Required Packages

This section documents the dependencies required by tlc-article. Package names are listed in alphabetical order. Search https://ctan.org/ for complete package descriptions and documentation.

Name	Description
appendix	The appendix package provides various ways of formatting the titles of appendices. Also (sub)appendices environments are provided that can be used, for example, for per chapter/section appendices.
array	An extended implementation of the array and tabular environments which extends the options for column formats, and provides 'programmable' format specifications.
bookmark	This package implements a new bookmark (outline) organization for package hyperref.
catchfile	This package catches the contents of a file and puts it in a macro.
csvsimple	The package provides a simple LaTeX interface for the processing of files with comma separated values (CSV); it relies on the key value syntax supported by pgfkeys to simplify usage.
enumitem	This package provides user control over the layout of the three basic list environments: enumerate, itemize and description.
fancyhdr	The package provides extensive facilities, both for constructing headers and footers, and for controlling their use (for example, at times when LaTeX would automatically change the heading style in use).
fontenc	The package allows the user to select font encodings, and for each encoding provides an interface to 'font-encoding specific' commands for each font.
geometry	The package provides an easy and flexible user interface to customize page layout, implementing autocentering and auto-balancing mechanisms so that the users have only to give the least description for the page layout.
glossaries	The glossaries package supports acronyms and multiple glossaries, and has provision for operation in several languages.
graphicx	The package builds upon the graphics package, providing a key-value interface for optional arguments to the 'includegraphics' command. This interface provides facilities that go far beyond what the graphics package offers on its own.
hyperref	The package is used to handle cross-referencing commands in LaTeX to produce hypertext links in the document.
ifthen ifthenelse, which can use a wide array of tests.	The package's basic command is

Name	Description
inputenc	The package translates various standard and other input encodings into a LaTeX internal language. The internal language is expressed entirely in TeXś base encoding (standard ASCII printable characters, carriage control tokens and TeX control sequences, the latter mostly defined by LaTeX).
lastpage	Reference the number of pages in your LaTeX document through the introduction of a new label which can be referenced with 'pagerefLastPage'.
listings	The package enables the user to typeset programs (programming code) within LaTeX; the source code is read directly by TeX – no frontend processor is needed.
lmodern	Latin modern fonts
longtable	Longtable allows you to write tables that continue to the next page. You can write captions within the table (typically at the start of the table), and headers and trailers for pages of table.
makecell	This package supports common layouts for tabular column heads in whole documents, based on one-column tabular environment.
multicol	Multicol defines a multicols environment which typesets text in multiple columns (up to a maximum of 10), and (by default) balances the end of each column at the end of the environment.
multirow	The package creates table cells that span multiple rows.
pdflscape	The package adds PDF support to the landscape environment of package lscape, by setting the PDF /Rotate page attribute.
pdfpages	This package simplifies the inclusion of external multipage PDF documents in LaTeX documents.
pgf-pie	This package provides the means to draw pie and variant charts using PGF/TikZ.
spverbatim	The spverbatim package provides an 'spverb' macro and an spverbatim environment that allow LaTeX to break lines at space characters.
tabularx	The package defines an environment tabularx, an extension of tabular which has an additional column designator, X, which creates a paragraph-like column whose width automatically expands so that the declared width of the environment is filled.
textcomp	The package supports the Text Companion fonts, which provide many text symbols (such as baht, bullet, copyright, musicalnote, onequarter, section, and yen), in the TS1 encoding.
titling	The titling package provides control over the typesetting of the 'maketitle' and 'thanks' commands, and makes the 'title', 'author' and 'date' information permanently available.
tocloft	Provides control over the typography of the Table of Contents, List of Figures and List of Tables, and the ability to create new 'List of ...' entries.
todonotes	The package lets the user mark things to do later, in a simple and visually appealing way. The package takes several options to enable customization / fine-tuning of the visual appearance.

Name	Description
xcolor	The package starts from the basic facilities of the color package, and provides easy driver-independent access to several kinds of color tints, shades, tones, and mixes of arbitrary colors.