tlc-article

Gary Allan Howard

May 17, 2019

Abstract

The tlc-article 'Getting Started Guide' covers how to install tlc-article both globally and locally, describes the general use case, how to customize your tlc-article environment, describes the commands tlc-article implements, and reveals the packages tlc-article depends upon.

Contents

1 Installation

This section describes how to install tlc-article either globally to make it available to your LaTeX environment or locally to the document you are authoring. And, this section identifies the prerequisites you must meet in order to clone a repository from GitHub.com and install software on your computer.

1.1 Prerequisites

The following prerequisites are needed.

Administrative privilege
: You will need administrative privileges to install tlc-article globally because 'sudo' is used.

SSH key
: You will need your private key to access GitHub.com. Please refer to `http://help.github.com/articles/generating-an-ssh-key` for instructions on 'Generating an SSH key'.

Enable your SSH key
: The following instructions enable your SSH key so you to not have to enter the passphrase for each git command.

1. eval $(ssh-agent -s)
2. ssh-add /.ssh/your-private-key
3. enter your passphrase

1.2 Local installation

A local installation is done by installing tlc-article into /the/path/to/your/document. Assuming your document is located at $HOME/mydoc the following shell commands will make tlc-article available to your document.

1. cd $HOME
2. git clone git@GitHub.com:Traap/tlc-article.git
3. cd tlc-article
4. mkdir $HOME/mydoc
5. cp -v tlc-article.cls $HOME/mydoc/.

1.3 Global installation

A global installation is done by installing tlc-article into your /path/to/your/texmf directory. Assuming a TexLive installation exists at $(kpsewhich -var-value TEXMFLOCAL) the following shell commands will make tlc-article available to your LaTeX environment.

1. cd $HOME
2. git clone git@GitHub.com:Traap/tlc-article.git
3. cd tlc-article
4. sudo mkdir -p $(kpsewhich -var-value TEXMFLOCAL)/tex/latex/tlc-article
5. sudo mv -v tlc-article.cls $(kpsewhich -var-value TEXMFLOCAL)/tex/latex/tlc-article/.
6. sudo mktexlsr $(kpsewhich -var-value TEXMFLOCAL)

Note You may remove your local installation by removing tlc-article.

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

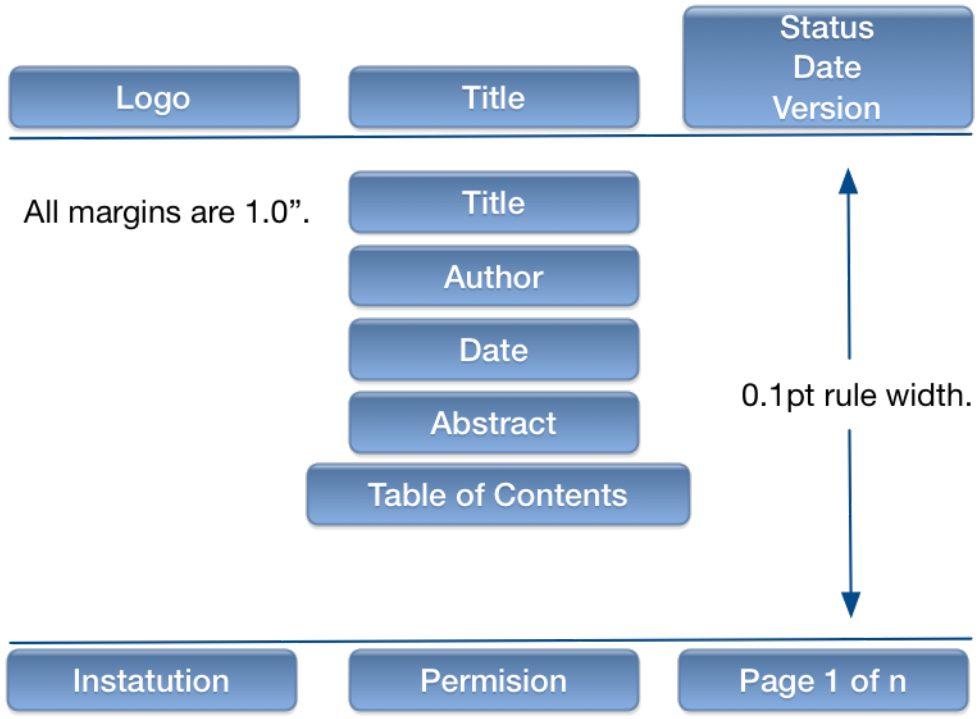

Figure 1: Document Layout

2.2 documentclass tlc-article

tlc-article extends the article document class. tlc-article provide 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, document abstract, and establish the Table of Contents. The sample below reveals how to use tlcTitlePageAndTableOfContents.

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

2.4 Table of Contents

The Table of Contents immediately follows the document abstract on page 1, uses dark blue for content, dots separate table of contents sections and page number, and uses roman numerals.

2.5 Header & Footer

fancyhdr is used to render the header and footer. The Author can override the tlc-article by providing an implementation in data/header-footer.tex or augment 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 ignore data/version.csv when data/header-footer.tex is defined.

Header

lhead When data/logo.png is found, logo.

chead Document Title

rhead When data/version.csv is present, status, date, and version columns.

Footer

lfoot When data/version.csv is present, institution column.

cfoot When data/version.csv is present, permission column.

rfoot Page 1 of N.

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 check for. tlc-article default implementation will be used when the file-hooks are now found.

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/additional-layout.tex tlc-article has a builtin header and footer strategy that is base on *fancyhdr*, *titling*, and *lastpage* LaTeX packages. The default implementation is show below:

```
\IfFileExists{\tlc@logoFile}
{ % Typeset the logo in the left side of the document header.
  \lhead{\includegraphics[width=3cm,height=1cm]{\tlc@logoFile}}
}
{% Else: no operation because tlc@logoFile does not exist.
}
% Typeset the title in the center of the document header.
\chead{\large{\thetitle}}
% Typeset version information in the right side of the document header.
\IfFileExists{\tlc@versionFile}
{
  % document status, document date and document version.
  \rhead{\tiny \tlc@status \\ \tlc@date \\ \tlc@version}
  % document owner.  This maybe a person or company name.
  \lfoot{\tiny \tlc@instatution}
  % document license. This maybe a license or word like confidential.
  \cfoot{\tiny \tlc@permission}
}
{% Else: no operation because tlc@versionFile does not exist.
}
\renewcommand{\headrulewidth}{0.1pt}
% eliminate head height too small warning, which is occurring because
% we are using multiple lines in our header.
\setlength\headheight{52pt}
% % footer applied to all pages.
\rfoot{\tiny{page \thepage~of~\pageref{LastPage}}}
\renewcommand{\footrulewidth}{0.1pt}
% we want our header and footer to remain consistent with a table of
% contents that span multiple pages.
\AtBeginDocument{\addtocontents{toc}{\protect\thispagestyle{fancy}}}
```

The default implementation can be overridden by defining data/header-footer.tex.

Note When data/header-footer.tex exists and is empty, your document will be typeset with the defaults from document-class article.

3.3 data/version.csv

tlc-article will populate the builtin header and footer with information extracted from data/version.csv when it is present. data/version.csv is a comma-separated-variable file that uses the pipe character as the field delimiter. data/version.csv uses the following column names:

version
: The version value is typeset in the rhead area. This field is used to convey the version the document was at the date it reached its current state.

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

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

instatution
: The institution value is typeset in the lfoot area. This field is used to tell the reader the author name or company name.

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

The exaction 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=\instatution,
    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@instatution{\tlcVersionPart{\version}}
  \def\tlc@permission{\tlcVersionPart{\version}}
}
{% Else: no operation because tlc@versionFile does not exist.
}
```

3.4 data/logo.png

tlc-article will typeset the lhead area with data/logo.png when it is present. Make sure your logo's height is not larger than 34pt to avoid 'Package Fancyhdr Warning: headheight is to small' warning.

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 rbg{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.

4.5 newcolumn type: L, C & R

New newcolumn type: L, C & R are Left, Center, and Right, respectively are designed to use with longtable. Data is wrapped within a table cell. The parameter defines the column width. As an example, L2cm yields a Left aligned, ragged right, wrapped text within a 2cm 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 should use when it becomes necessary to use packages not provided by tlc-article and to design commands that are specific to your document.

4.7 data/header-footer.tex

data/header-footer.tex is an architectural hook the Author should use to completely override the document layout tlc-article implements.

4.8 data/version.csv

data/version.csv is by used 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 define. 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.csvin your data/header-footer.tex.

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

4.9 data/logo.png

data/logo.png is used to place your logo in the header created by tlc-article.

5 Required Packages

This section documents the dependencies of the required package tlc-article has. Package names are listed in alphabetical order. A complete description of each package is found at http://www.ctan.org/. At this writing, you can type in the package name and press the search button to learn more about each package.

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.
babel	This package manages culturally-determined typographical (and other) rules for a wide range of languages.
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.
fontenc	The package alows 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 auto-centering and auto-balancing mechanisms so that the users have only to give the least description for the page layout.
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 hyperref package is used to handle cross-referencing commands in LaTeX to produce hypertext links in the document.

Name	Description
hyperref	The package is used to handle cross-referencing commands in LaTeX to produce hypertext links in the document.
inputenc	The package translates various standard and other input encodings into a LaTeX internal language. The internal langage is expressed entierly in TeXś base encoding (standard ASCII printable characters, carriage control tokes and TeX control sequences, the later mostly defined by LaTeX).
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).
jancyhdr	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).
lastpage	Reference the number of pages in your LaTeX document through the introduction of a new label which can be referenced like 'gpagerefLastPage' to give a reference to the last page of a document.
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.
parskip	Simply changing 'gparskip' and 'parindent' leaves a layout that is untidy; this package (though it is no substitute for a properly-designed class) helps alleviate this untidiness.
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.
pdf-pie	Ths package provides the means to draw pie (and variant charts) using PGF/TikZ.
spverbatim	The spverbatim package provides an 'gspverb' macro that is analogous to 'verb' and an spverbatim environment that is analogous to verbatim with the difference being that 'spverb' and spverbatim allow LaTeX to break lines at space characters.

Name	Description
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 'gmaketitle' command 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 ...'. The ToC 'gparskip' may be changed.
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.
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.