

# L<sup>A</sup>T<sub>E</sub>X Class `huawei`

## User's Guide

Huawei Technologies Co., Ltd.

YEGOR BUGAYENKO

Version: `0.6.1`

Date: `2021/06/15`

## 1 Overview

The provided class `huawei` helps you design your work documents and presentations keeping the code short and the style elegant enough both for management and technical papers. To use the class you simply mention its name in the preamble:

```
\documentclass{huawei}
\begin{document}
Hello, world!
\end{document}
```

The document rendered from this L<sup>A</sup>T<sub>E</sub>X code will look similar to the document you are reading now. We recommend you to use `latexmk` to compile your `.tex` files to `.pdf`. The simplest setup will require a few files staying next to your `.tex` file, in the same directory (“story” is the name of your project here):

```
story\
  .latexmkrc
  .gitignore
  Makefile
  story.tex
```

The content of your `.latexmkrc` file would be this:

```
$pdflatex = 'pdflatex %0 --shell-escape %S';
```

Make sure the `.gitignore` file lists all the files generated by `pdflatex` during the compilation. You don't need to commit them to your repository, since they are temporary and will be generated again when you compile your document.

The recommended content of the `Makefile` would be this:

```
TEXS=$(wildcard *.tex)
PDFS=$(TEXS:.tex=.pdf)
all: $(PDFS)
%.pdf: %.tex
    latexmk -pdf $<
```

In order to compile the document, just say `make` on the command line.

## 2 Layout Options

There are a few class options, provided in square brackets after the `\documentclass`, which can help you fine-tune the layout of your document:

`landscape` makes the document in landscape format, also changing the size of the paper to 16x9 inches (the default page size is A4), making it perfect for presentations.

`nocover` avoid printing the cover images on the first page by the `\PrintTitlePage` command.

`anonymous` removes the name of the author everywhere, including the bottom of the page, where the author's name stays next to the name of the company.

`nobrand` avoid mentioning the brand of Huawei anywhere in the document and removes the logo too.

`nosecurity` avoids mentioning the level of security at the right top corner of the document and also avoids showing the ID of the author where it usually is visible.

`nodate` don't show the date and time at the bottom of each page, where they usually are rendered in ISO 8601 format.

`nopaging` avoids page numbers at the bottom of each page.

## 3 Preamble

In the preamble you can specify meta information about the document, such as its title or author's name, here is how:

```
\documentclass{huawei}
\renewcommand*\thetitle{Making Compression 15% Faster}
\renewcommand*\thesubtitle{Technical Report}
\renewcommand*\theauthor{Yegor Bugayenko}
\begin{document}
```

```
\maketitle
Hello, world!
\end{document}
```

It's recommended to use `\renewcommand*` instead of `\renewcommand` in order to let  $\TeX$  catch you if by mistake a new line gets into the content.

The following meta commands are defined:

`\thetitle` is the main title of the document to be used in the text and in the properties of the PDF document.

`\thesubtitle` is the subtitle to be printed under the title.

`\theauthor` is the author of the document in "first-name last-name" format.

`\theid` is the internal ID of the author, if it's applicable.

`\thesecurity` is the level of security of the document, which is usually printed at the top right corner of it; usual values are "Internal," "Confidential," or "Secret."

Default values of all these commands are empty. If you don't renew them in your document, nothing will be printed.

## 4 Custom Commands

There is a number of supplementary commands for better text formatting, which we introduced:

`\ff{text}` makes the text fixed-font with a nice border around.

`\tbd{text}` highlights the text, which is expected to be improved later (tbd stands for "To Be Determined"), like [this one](#).

Inside the document body you can use these commands:

`\PrintFirstPage{front-image}` prints the first page of a project charter or a similar landscape documents, placing the image `front-image.pdf` on the front (the file should be present in the current dir. If you don't have the front image file, just leave the first argument empty).

`\PrintLastPage{}` prints the last page of a project charter or a similar landscape document.

`\PrintThankYouPage{}` prints the last page with a "Thank You" message in the center.

`\PrintDisclaimer{}` prints a paragraph at the bottom of the page with a standard disclaimer.

## 5 Best Practices

You are free to design your documents any way you want. However, it would be convenient for yourself and for your readers, if you follow the convention we have for business and technical documents. The `samples` directory contains a number of sample documents, which we suggest you to use as templates when you start making new documents.

The rule of thumb is simple: try *not* to format your documents. Instead, let the class designed by us do this work for you. Just type the content without changing the layout, adding colors, changing fonts, etc. The less you modify the look-and-feel, the better your documents will be perceived by your readers.

### 5.1 Two Columns

In the landscape format it's recommended to use two columns, for better readability of the text. Here is how:

```
\documentclass{huawei}
\begin{document}
\newpage
\begin{multicols}{2}
\section*{First}
Here goes the first column content.
\columnbreak
\section*{Second}
Here goes the second column content.
\end{multicols}
\end{document}
```

A more complete example is in the `samples/huawei-charter.tex`.

### 5.2 Crumbs

When you need to put many small information pieces into one page, we recommend you to use “crumbs”:

```
\documentclass{huawei}
\begin{document}
\newpage
\section*{Project Details}
\begin{multicols}{2}
```

```
\raggedright
\crumb{Budget}{\$100K}

\crumb{Duration}{5 months}
\end{multicols}
\end{document}
```

A more complete example is in the `samples/huawei-charter.tex`.

### 5.3 Code Blocks

When you need to show some source code, we recommend to use `minted` environment, for example:

```
\begin{minted}{text}
void foo() {
    return "Hello, world!";
}
\end{minted}
```

This is what you will see:

```
void foo() {
    return "Hello, world!";
}
```