

TeXBlend – Compile Segments of LaTeX Documents

Michal Hoftich*

Version v0.1, 2023-11-24

Homepage: <https://www.kodymirus.cz/texblend/>
Issue tracker: <https://github.com/michal-h21/texblend>

Contents

1	Introduction	1
2	Structure of TeX Files	2
2.1	Placement of the Main Document and Templates	3
2.2	Custom Variables in Templates	3
3	Usage	4
3.1	Command Line Options	4
3.2	Compilation Options	5
3.3	Usage of Templates	5
3.4	HTML Support	5
4	License	6
5	Changelog	6

1 Introduction

This tool compiles individual files that are included as parts of larger documents. It utilizes the preamble of the main document but disregards all other included files.

The main purpose is to allow fast compilation of particular chapters or sections, eliminating the need to recompile the entire document. This facilitates an efficient way to check for formatting or syntax errors in the particular part of the document being worked on.

*<michal.h21@gmail.com>

2 Structure of \TeX Files

The basic usage is as follows: consider a main document that includes files for individual chapters, named for example `main.tex`:

```
\documentclass{article}
\author{John Doe}
\title{Sample Document}
\usepackage{upquote}
\usepackage{microtype}
\usepackage{hyperref}
\begin{document}
\maketitle
\tableofcontents
\input{intro}
\input{usage}
\end{document}
```

We can compile this document to obtain a PDF file containing the text of all chapters. However, by using \TeX Blend, we can generate separate PDFs for each chapter. The advantage is that all commands defined in the packages used in the main document will be available.

In the included files, we can use special directives that indicate the main file and the engine to be used for compilation. Similar directives are employed by various \TeX editors, such as \TeX Shop or \TeX Works.

For instance, a file like `intro.tex`, which is included in the main document, might look like this:

```
% !TEX root = main.tex
% !TEX TS-program = lualatex

\section{Introduction}
This tool compiles individual files that are included
as parts of larger documents.
```

Notice the comments at the beginning of the file. These are directives, indicating the main document and the engine to be used for compilation. Many \TeX editors support these directives, and we utilize them in \TeX Blend to effectively control the compilation process too.

These directives follow a specific format, requiring placement at the beginning of the file and commencing with a percentage sign. Subsequently, there is a space followed by either the marker `!TEX` or `!TeX`, both of which are recognized formats. The next element is the property name with an optional `TS-` prefix (in line with the \TeX Shop editor), followed by an equals sign and, finally, the value.

Two crucial properties are `root`, indicating the name of the main file, and `program`, through which we can select the program for compiling the document. However, it is possible to specify additional properties, which can then be utilized in template documents.

2.1 Placement of the Main Document and Templates

You can specify the full path to the main document in the root property. `TeXBlend` can also be used to compile documents using a universal template. The path to this template can be specified in full or can leverage the directory structure of TeX by placing it in a directory recognized by the KPSE library. This directory is typically designated for user packages, commonly found at `~/texmf/tex/latex`, although it may vary on different systems. You can identify its location on your system using the command:

```
$ kpsewhich --var-value TEXMFHOME
```

This command typically outputs a path similar to this: `/home/username/texmf`. If this directory does not exist, create it. Further, within this directory, create the subdirectories `tex/latex/templates`:

```
$ mkdir -p ~/texmf/tex/latex/templates
```

You can then place your templates in this directory. To test the availability of the template, run the following command:

```
$ kpsewhich templatename.tex
```

It should print path of the template document.

2.2 Custom Variables in Templates

As templates are intended to be universal across various projects, it is possible to incorporate variables into them. These variables can be populated from properties in the directives, such as the document title.

To include variables for expansion in the template, you can use the notation `{{variableName}}`. For example, to set the document title, you can use `\title{{title}}`. The double curly braces around the variable will be replaced with its value, while the third pair of curly braces will remain in the document.

Say that you have a following template saved in `~/texmf/tex/latex/templates/mytemplate.tex`:

```
\documentclass{article}
\title{{title}}
\author{Michal}
\usepackage{microtype}
\begin{document}
\end{document}
```

A document that utilizes this template may look like the following:

```
% !TEX root = mytemplate.tex
% !TEX title = hello world
\maketitle
```

This produces a following document with variables expanded:

```
\documentclass{article}
\title{hello world}
\author{Michal}
\usepackage{microtype}
\begin{document}
% !TEX root = mytemplate.tex
% !TEX title = hello world
\maketitle
\end{document}
```

You can require expansion of variables using the `--expand` command line option.

3 Usage

The basic usage of `TeXBlend` is as follows:

```
$ texblend chapter.tex
```

This command combines the text from the document `chapter.tex` with the preamble from the main document specified in the root directive and directly compiles the resulting document. The output will be saved as `chapter.pdf`.

3.1 Command Line Options

`TeXBlend` supports a variety of command-line options:

```
texblend [options] filename
```

Available options:

<code>-e,--expand</code>	Expand variables in the template
<code>-h,--help</code>	Print the help message
<code>-H,--HTML</code>	Compile to HTML using TeX4ht
<code>-l,--lualatex</code>	Select lualatex as compiler
<code>-o,--output (default "")</code>	Set the output file name
<code>-O,--texoptions (default "")</code>	Extra options that should be passed to the LaTeX compiler
<code>-p,--pdflatex</code>	Select pdflatex as compiler
<code>-P,--print</code>	Don't compile the document, print the expanded template instead
<code>-s,--shell-escape</code>	Enable execution of external commands in LaTeX
<code>-t,--template (default "")</code>	Use explicit template instead of the one provided in the input file
<code>-v,--version</code>	Print version info

<code>-x,--xelatex</code>	Select xelatex as compiler
<code><filename> (string)</code>	Use "-" if you want to pass document from the standard input

You can join multiple short options, for example the following call uses separate short options:

```
$ texblend -e -l -o test chapter.tex
```

It can be shortened to this form:

```
$ texblend -elo test chapter.tex
```

3.2 Compilation Options

You can override the program directive, which specifies the \TeX engine used, with one of the options `--lualatex`, `--xelatex`, or `--pdflatex`. The default program used when the document lacks a program directive and none of these options is used is `pdflatex`.

If the document uses external commands, such as with the `Minted` or `Imakeidx` packages, you can utilize the `--shell-escape` option.

You can specify additional options for the LaTeX compiler using the `--texoptions` option.

Instead of compiling, you can print the assembled document to standard output using the option `--print`.

To change name of the generated PDF file, use the `--output` option.

3.3 Usage of Templates

Instead of the main document specified in the `root` directive, you can use another document as a template, chosen using the `--template` option. You can use the entire file path or simply specify the file name, making use of the `kpsewhich` command (see section 2.1) to locate the file.

If you are using variables in the template (see section 2.2), use the `--expand` option to expand them with the values specified in the document's directives.

3.4 HTML Support

\TeX Blend supports direct export to HTML using \TeX 4ht. A note regarding the `--texoptions` command line option: you can utilize this option to pass all command-line arguments to `make4ht`. Ensure to include the `-` character in place of the filename for proper execution.

For example, if you want to use the `fn-in` option for endnotes, use the following command:

```
$ texblend -HO '- "fn-in"' chapter.tex
```

If you want to use a configuration file in addition, you will need this command:

```
$ texblend -HO '-c config.cfg - "fn-in"' chapter.tex
```

4 License

Permission is granted to copy, distribute and/or modify this software under the terms of the \LaTeX Project Public License, version 1.3.

5 Changelog

2023-11-24 Initial release