

T_EX4ht configurations

Michal Hoftich

August 13, 2021

Charles University, Faculty of Education, Library

Introduction

Process Overview

make4ht

T_EX4ht configuration

Troubleshooting

Introduction

Basic Information

- T_EX4ht converts L^AT_EX files to HTML, ODT, Docbook, Tei and other formats
- It uses L^AT_EX itself for the conversion, so it supports custom commands and most packages.
- It supports basic features, like font sizes, weights and style out of the box, but if we want to support more advanced features, like sections, footnotes, tables or images, we must provide configuration files.
- As any L^AT_EX package can redefine L^AT_EX core or other package's code, conflicts can happen.

Process Overview

- TeX4ht code is loaded using the `tex4ht.sty` package.
- It loads all configuration files for the supported packages and inserts XML tags.
- Tags are passed in `\special` commands to the DVI file.

The `tex4ht` command

- The DVI file is processed by the `tex4ht` command, that produces XHML or HTML files and two special files:
 - The `.lg` file contains instructions for CSS or generated picture file names.
 - The `.idv` file is special version of the DVI file that contains only parts of the original DVI file that should be transformed to pictures.

The t4ht command

- It creates the CSS file and creates pictures.

make4ht

- T_EX4ht provided several scripts to make the conversion easier, but they were obsoleted by two tools:
 - tex4ebook converts L^AT_EX to Epub and Mobi formats.
 - make4ht converts to all other supported formats.
- both have similar features and options.

- Unicode output by default.
- It is possible to select Lua \TeX or Xe \TeX for the compilation.
- Easy selection of the output format.
- Configurable level of terminal output.
- Extension support.
- Post-processing of the generated files.
- Lua build file support.

```
$ make4ht [options] filename.tex "[tex4ht options]"
```

or

```
$ tex4ebook [options] filename.tex \  
"[tex4ht options]"
```

Engine selection

Compile file using Lua \LaTeX :

```
$ make4ht -l filename.tex
```

Compile file using Xe \LaTeX :

```
$ make4ht -x filename.tex
```

Select output format

Convert to ODT:

```
$ make4ht -f odt filename.tex
```

Convert to HTML 5 (default):

```
$ make4ht -f html5 filename.tex
```

Convert to Epub 3:

```
$ tex4ebook -f epub3 filename.tex
```

make4ht compiles \TeX file three times by default. To make the compilation faster, use the **draft** mode:

```
$ make4ht -m draft filename.tex
```

To clean all temporary files, as well as generated XML files and pictures, use:

```
$ make4ht -m clean filename.tex
```

make4ht hides most of the terminal output by default. It only shows errors and warnings. To show all messages, use:

```
$ make4ht -a debug filename.tex
```


Extensions

Extensions can be enabled and disabled using the following syntax:

```
# enable extension
$ make4ht -f html5+<extension_name> filename.tex
# disable extension
$ make4ht -f html5-<extension_name> filename.tex
```

One extension `common_domfilters`, is used by default. It fixes common issues in the generated HTML and XML files. To disable it, use:

```
$ make4ht -f html5-common_domfilters filename.tex
```

To compile RTeX documents (you need Knitr installed):

```
$ make4ht -f html5+preprocess_input filename.rtex
```

List of supported extensions can be found in the make4ht documentation.

- Build files can be used to change the compilation sequence.
- They can define post-processing filters (Lua functions on external commands).
- It is also possible to specify command for the picture generation (like `dvisvgm` or `dvipng`).

Sample build file

```
-- support the fast draft mode
if mode=="draft" then
  Make:htlatex{}
else
  Make:htlatex{}
  -- create index page using the xindex command
  Make:xindex {}
  Make:htlatex{}
  Make:htlatex{}
end
```

Use build file

- The previous sample file can be used to support index in the HTML file.
- The **mode** variable holds value of the **-m** option.
- Commands are executed using **Make:command_name{options}**.

To pass the build file to make4ht use:

```
$ make4ht -e build.lua filename.tex
```

T_EX4ht configuration

T_EX4ht is highly configurable. The easiest way how to change it's features is using options. They can be required using:

```
$ make4ht filename.tex "option1,option2,..."
```

Useful options

Each output format has different set of options. Some useful options for HTML include:

fn-in put footnotes at the bottom of the current page instead of standalone pages.

svg create SVG pictures.

2 put each section to a separate HTML page.

sec-filenames name HTML files for sections according to the section title.

info output description of the basic configurations to the .log file.

Configuration files

- Configuration files can pass the options.
- They can contain the CSS code.
- They can be used to configure the XML tags.

They can be required using:

```
$ make4ht -c config.cfg filename.tex
```

Basic structure of the configuration file

```
\Preamble{xhtml,options}  
% early configurations  
\begin{document}  
% configurations that should appear  
% after \begin{document}  
\EndPreamble
```

- `\Preamble` can pass the options. `xhtml` option needs to be always first, even if no other option is used.
- `TEX4ht` redefines all commands after the document preamble, but some code may be executed later. This can be overridden by configurations put after the `\begin{document}`.

Sample configuration file

The following configuration file can be used to create HTML files where sections are in separate files named after the section titles, with footnotes at the end of the each file where they appear.

CSS is used to set the maximum width of the page.

```
\Preamble{xhtml,2,sec-filename,fn-in}  
\Css{body{max-width:60ch;margin:1em auto;}}  
\begin{document}  
\EndPreamble
```

Require external CSS file

In this example we use the LaTeX.css style to change the document appearance completely:

```
\Preamble{xhtml,2,sec-filename,fn-in}
\Configure{@HEAD}{
  \HCode{
    <link rel="stylesheet"
    href="https://latex.now.sh/style.css">
  \Hnewline
  }
}
\begin{document}
\EndPreamble
```

Commands useful in the configuration files

- The `\Configure` command is used to insert code to hooks that were declared by configuration files for used packages and \LaTeX core.
- The `\HCode` command is used to insert XML tags.
- The `\Css` command is used for shorter CSS code.
- The `\Hnewline` command is used to output a line break to the XML file.

Math appearance

By default, math output is mixture of HTML elements and pictures. It doesn't look well. It is better to use the SVG graphics for pictures and require pictorial math also for inline math:

```
$ make4ht filename.tex "pic-m,svg"
```

You can use the `dvisvgm_hashes` extension for faster compilation of documents that produce lot of images. It can use multiple processes and compiles only pictures that changed from the previous run:

```
$ make4ht -f html5+dvisvgm_hashes filename.tex \  
"svg,pic-m"
```

MathML is a better solution than pictures in general. The downside is that it doesn't work in all web browsers:

```
make4ht filename.tex "mathml"
```

You can use the MathJax library to require MathML support in all browsers:

```
make4ht filename.tex "mathml,mathjax"
```


You can also let MathJax to completely handle math:

```
make4ht filename.tex "mathjax"
```

\TeX 4ht doesn't expand math contents in this case, so you will need to pass definitions of custom commands to MathJax.

Pass configuration to MathJax

```
\Preamble{xhtml,mathjax}
\Configure{MathJaxConfig}{
  tex: {
    \detokenize{%
      macros: {
        mycmd: "\\sin{a}"
      }
    }
  }
}
\begin{document}
\EndPreamble
```

T_EX4ht needs to know image dimensions to correctly set them in the HTML file. It uses a .xbb file for that. It can be created using:

```
ebb -x imagename.jpg
```

Troubleshooting

Package conflicts

- $\text{T}_{\text{E}}\text{X}4\text{ht}$ needs to redefine lot of internal \LaTeX commands, as well as commands from various packages. It may sometimes lead to package clashes, if other packages redefine the same commands.
- Some packages don't work well with $\text{T}_{\text{E}}\text{X}4\text{ht}$, because they require the PDF mode for example.
- It is ideal to run $\text{T}_{\text{E}}\text{X}4\text{ht}$ after you add any new package, to find if it fails.
- Please let us know about conflicts at $\text{T}_{\text{E}}\text{X}4\text{ht}$ mailing list, issue tracker, or at [TeX.sx](https://www.tex.sx) (use the *tex4ht* tag).

First aid for the package conflict

We want to avoid need to any changes in your \TeX documents for the \TeX 4ht support, but it is sometimes the easiest way how to handle the package conflict.

Use the `\ifdefined\HCode` command to detect \TeX 4ht, and avoid loading of the conflicting packages using that:

```
\documentclass{article}
...
\ifdefined\HCode
\usepackage{graphicx}
\else
\usepackage[pdftex]{graphicx}
\fi
...
```

T_EX4ht homepage

tug.org/tex4ht/

New T_EX4ht documentation

www.kodymirus.cz/tex4ht-doc/tex4ht-doc.html

T_EX4ht options

[https://www.kodymirus.cz/tex4ht-doc/](https://www.kodymirus.cz/tex4ht-doc/texfourhtOptions.html)

[texfourhtOptions.html](https://www.kodymirus.cz/tex4ht-doc/texfourhtOptions.html)

Make4ht documentation

www.kodymirus.cz/make4ht/make4ht-doc.html

Issue tracker

puszcza.gnu.org.ua/bugs/?group=tex4ht