

Faith, Hearing, and the Power of Music in Devotional Music of the Spanish Empire

Andrew A. Cashner

Sources

The main text is written in Pandoc's markdown in the files in the `chapters` directory. Configuration for Pandoc and LaTeX (used by Pandoc) is in the `config` directory. There are also LaTeX class and package files in the main directory used to process the full document (`aacbook.sty`) and the float files (`poemtranslation.sty` for poem and translation in parallel columns, and others). Citations are processed by Pandoc with the `pandoc-citeproc` filter, using the BibLaTeX database `master.bib`.

The other directories (`music-examples`, `figures`, `tables`, `poem-examples`) contain individual Markdown files with the caption for each floating item to be inserted in the main text. Within each of these directories, the `src` directory contains the LaTeX or Lilypond source file of the actual float. Those files will be processed separately and included as images. The processed images are in their own directory tree under `img/`.

Cross-references

Cross-references are resolved by a preprocessing step, using my Guile module `xref`. Just as in LaTeX, a *label* identifies an item to be numbered and a *ref* references the number of that item. Different from LaTeX, each label has both a type and a key. The format is always `label:type:key`: for example, the introductory chapter might be `label:ch:intro` and a table of names might be `label:table:names`. A reference to these would be `ref:ch:intro` and `ref:table:names`.

The (`xref reflabels`) module makes a list of all the label types and corresponding keys and then replaces the references with the appropriate number, based on the ordering of the labels within the source. The module (`xref listfiles`) converts the label list into a list of filenames to be included. It expects a configuration file, written as a Scheme association list, specifying which labels should be converted to filenames and how (e.g., `label:fig:face` is converted to `figures/face.md`; but `label:ch:intro` is not a float and can be ignored).

Per publisher specifications, the system numbers the floats consecutively by type throughout the whole document (not chapterwise as in LaTeX). They are included at the end of the PDF document and as a separate ODT file.

Compilation

Requirements

To create a PDF or ODT document from the sources in Markdown, LaTeX, and Lilypond, the following free software is needed:

- To compile complete document
 - GNU Bash or other Unix-like shell, Version 4.4.19 or higher
 - GNU Make, version 4.2.1 or higher
- To resolve cross-references
 - GNU Guile, version 2.0.14 or higher,
- To convert text in Markdown to PDF and ODT, and compile tables and poems
 - Pandoc, version 1.19 or higher
 - LaTeX full TeXLive installation, 2017 or later
- To compile musical examples
 - Lilypond, version 2.19.81 or higher

The shell and Make are only needed to run the Makefile. It would be possible to compile the whole document just using Guile (just load the Guile modules and invoke the functions as in the Makefile).

Printing the plain text source

In an emergency, the Markdown files, encoded in UTF-8, can be printed directly from the source files in plain text and will be perfectly legible. With the utilities `paps` and `ps2pdf` you can convert them to PDF, for example:

```
cat chapters/chapter-*.md | paps --paper letter --cpi 12 | ps2pdf - out.pdf
```

The book has been produced using Fedora 27. It was written using Vim.

Building the document and subfiles

The `Makefile` calls the Guile scripts `make-floats.scm` and `make-text.scm` in the `scripts` directory. These in turn use my Guile modules `shell` and `xref`.

- To clean auxiliary files: `make clean`
- To compile all float files as PDFs for inclusion in the master document: `make floats`
 - This produces PDF files for each of the float files in `tables`, `poetry-examples`, and `music-examples`. For tables and poems, it compiles them with LaTeX; for music-examples, with Lilypond. For tables and music examples, it crops them.
- To convert the whole text: `make`
 - This converts the whole text, including floats in order of their labels in the main text, to a single PDF
 - It also creates separate ODT files for each chapter and one that includes all the floats in a single ODT