Contents

1 Identity2

1.1 General rendering notes

The specific of how this document is produced are in the ”code” section

1.1.1 Rendering Ascii/Unicode Art

www Convert ascii to image

www App, aafigure

1.1.2 eBooks - ePub

The basic structure of a ePub book is as follows, I have created eBook, mobi etc entirely from python scripts and it is possible and relatively easy. In practice however I found it easier to roughly assemble the book by script and through the editing stages, however instead of programming for all the edge cases just open it in **Sigil** to finalize, test, audit and final render.

ePub2 hello world

```
.
├── mimetype                (file) THIS ONE MUST BE UNCOMPRESSED
├── META-INF
│   └── container.xml       (file)
└── OEBPS
```

```
├── book.ncx              (file)
├── book.opf              (file)
└── chapter01.xhtml       (file)
```

The file container.xml serves as a ”bootstrap” for the book.

The .opf file contains at a minimum, metadata of the book, the list of all files (except for mimetype, container.xml, and the .opf) that compose the book, with their full path, IDs and media-types, and, lastly, the ”linear” list of topics in which the book should be read.

The content of the dc:identifier element of the .opf file has to match the property of the meta element with name dtb:uid. Otherwise, **www** Epub-validate

does not validate the document.

Testing suite: Kindle Previewer 2, Adobe Digital Editions, iBooks, then on-device with Kindle Fire, Paperwhite, iPad and Android.

Font: serif sans-serif cursive fantasy monospace