Fountain Tools for LibreOffice

Copyright 2014

by

Richard Alexander Hall

http://earthbound.io/contact

For license details, please see README.md.

GitHub repositories:

https://github.com/r-alex-hall/FountainLibreOfficeTools

https://github.com/r-alex-hall/CourierMegaFont

Table of Contents

General introduction

See README.md for an overview of the components of this toolset.

The document which you now read explains how to install and use this toolset to create industry-standard formatted screenplays from fountain plain-text source documents, via LibreOffice. This also introduces how to write fountain plain text screenplays for this toolset to convert.

For an introduction to fountain, skip to the next section (depending on where you may be reading this document, you can e.g. go back one page to the “Table of Contents,” and CTRL+click, click or double-click the “Introduction to fountain” section link, to jump there.

To jump to how to set up and then use this toolset, skip to the sections labeled “Installation” and “Usage.”

Workflow overview

> *Note: .odt = Open Document Text (an open-standard, cross-platform document format, which LibreOffice and many other office softwares use).*

This toolset provides the following workflows. Note that many professional screenwriting programs that are “worth their salt” don't need any kind of intermediary format: you can just import fountain directly to them. The worfklow:

- Plain-text (.txt or .fountain) fountain screenplay source file →
  - .odt (LibreOffice Writer Open Document format file), styled and formatted to industry standards via included styles. The resultant document may be exported to a universally readable .pdf file. Then, supposing said script goes into production: →
    - .rtf (Microsoft's Rich Text Format, via LibreOffice) →
      - any professional screenwriting program.
- Alternate path from fountain: .odt file formatted for ebook export →
  - .doc (Microsoft Word) export →
    - Conversion to many different ebook formats for electronic publication, e.g. via smashwords.com

Introduction to Fountain

Fountain is screenwriting in plain text. It allows you to work on your screenplay anywhere, on any computer (or device), with any software that can edit and save plain text, such as Windows Notepad, Mac SimpleText/TextEdit, or any text editor on Unix/Gnu/Linux.

(If you're curious, I personally prefer the cross-platform Notepad++, with a theme called Solarized Dark (or Light, depending on lighting conditions), using DejaVu Sans Mono 14pt.)

You can even use smart phone SMS/text messages or text editors, if the text can be made accessible

to LibreOffice and the alt.search Replace extension.

Even as plain text, fountain markup conveys a sense that it is a screenplay. Moreover, because it’s plain text (which is accessible on any modern computing device, and therby essentially universal) it's also a great format to write and archive screenplays without worry of file-format obsolescence or incompatibility; it is in other words the ultimate portable and future-proof format.

The provided .odt template supports a subset of fountain/sreenplay elements, which subset is in my opinion all that any spec screenwriter really needs. You may need additional elemements for production, but those will be provided by the "big boy" tools which you (should) use in production. The following very short fountain screenplay demonstrates the supported subset (minus parentheticals and page breaks), which you may also see formatted to a proper screenplay in the file:

"LibreOffice_ScreenplayTemplate_v1.3.4.2 (standard).odt":

The super-short fountain screenplay:

```
FADE IN:

EXT. SPACE--NIGHT

Light pinkish-violet and blue nebulae drift over an ocean
of glittery, colorful stars.

GOD
Let there be light!

**NOTE: What comes to your mind? Write it!**

> THE BEGINNING <
```

You see, then, that the subset my template uses includes these screenplay elements:

- Transition
- Scene slug
- Action
- Character slug (parentheticals also supported but not shown)
- Dialogue
- Local text formatting, with all combinations of the following styles supported:
  - italics
  - bold

    - underline
- Centering layout

Follows examples of text formatting in fountain; to combine them, you nest asterisks and/or underscores:

```
*italic*

**bold**

***bold italic***

_underlined_
```

As you saw in the example, to center text you surround it with angle brackets (angled the inverse of e.g. what you use for HTML tags:

```
>THE END<
```

To combine text formats, nest asterisks inside underscores, or in other words surround the asterisks with underscores. Perhaps the formal fountain standard will tolerate underscored within asterisks, but my format batches won't;

Incorrect usage:

~~*_Underscored, italicized text_*.~~

Correct usage:

```
_*Underscored, italicized text*_.
```

More formatting detail than you may need appears in the final section: “Technical notes and nits,” which includes a practical caution about excessive use of ALL CAPS in screenplays, and other unnecessary elements and (erred) conventions.

Installation

> *Note that these instructions often have page breaks after large images. Just read through to the next.*

LibreOffice

First, install LibreOffice (if you don't already have it). You can download it from:

http://libreoffice.org

Alt.SearchReplace extension

Then, install the extension this toolset relies on, from this file amidst this distribution:

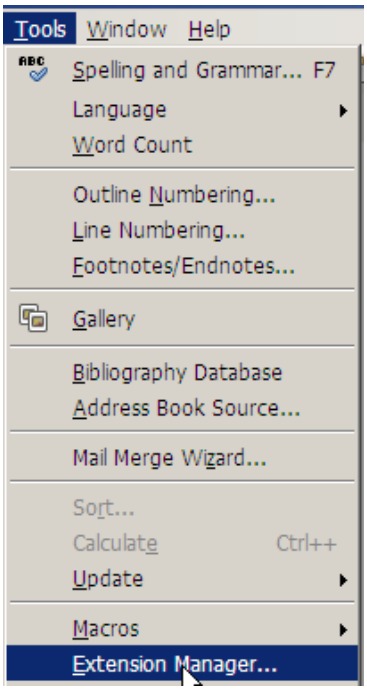
libreOfficeExtension_AltSearch-1.4.oxt

Or from:

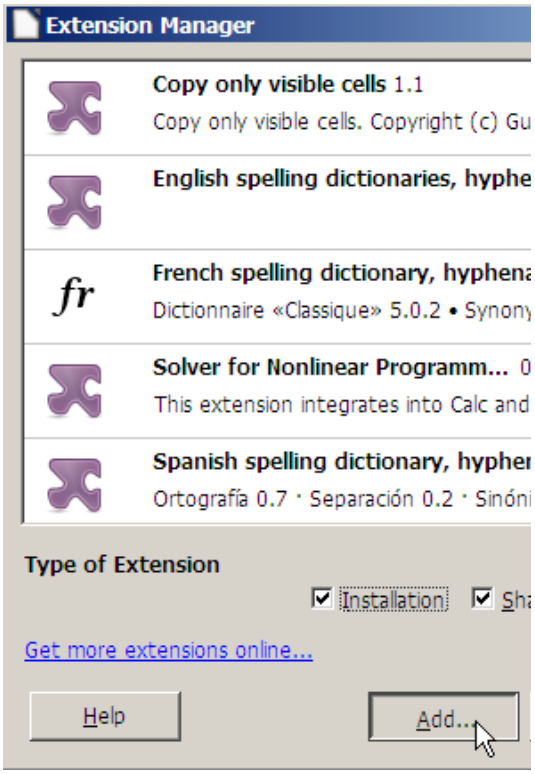
http://extensions.libreoffice.org/extension-center/alternative-dialog-find-replace-for-writer

You may install it by launching LibreOffice Writer, then:

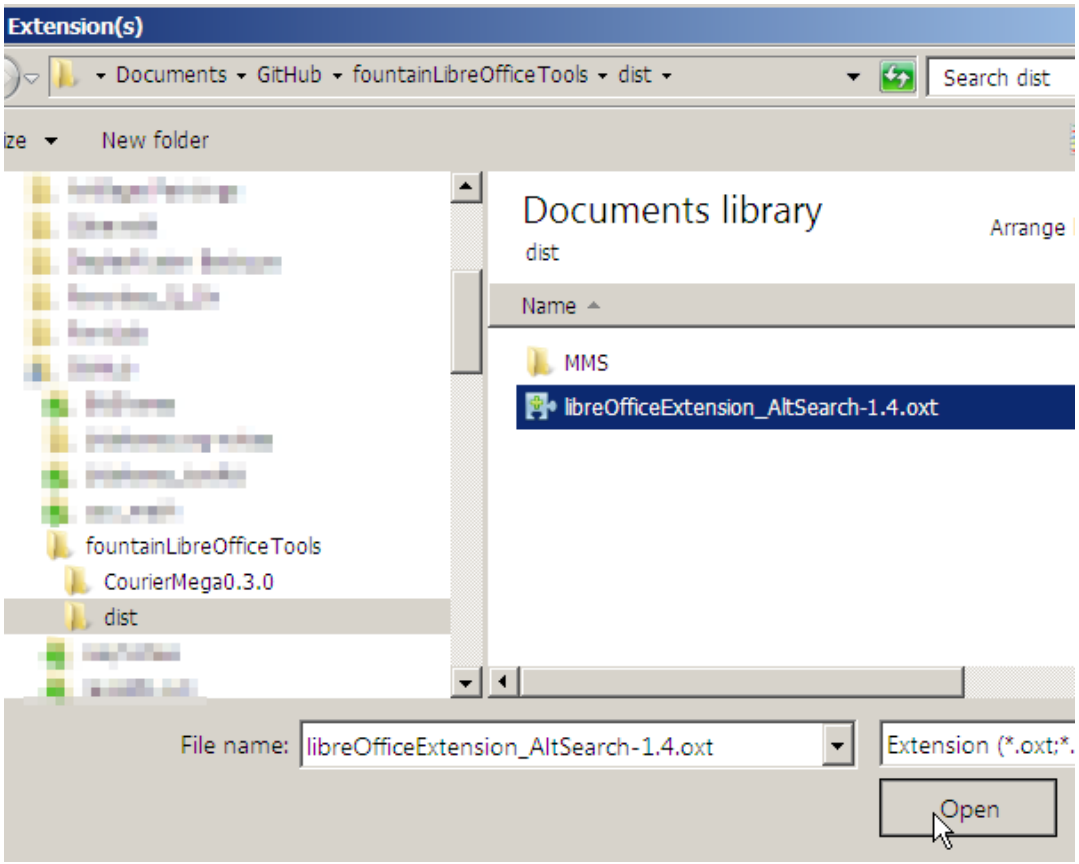
Go to the *Tools menu → extension manager* . . .



Click "Add..." . . .



Navigate to the file, select it, and click "Open" . . .



Install it as prompted, as you prefer (for yourself or every user). You may have to restart LibreOffice.

Extension batch script

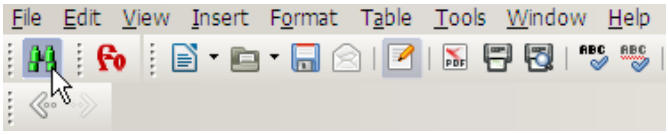
Now install the batch script I provide for this extension.

Open the following file from this distribution (it may be in a subfolder) in a plain text editor:

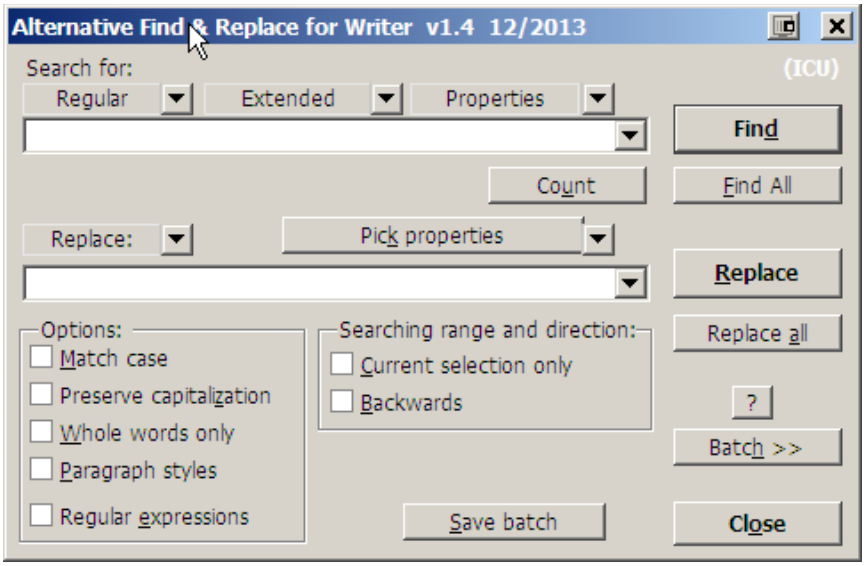
AltSearchScriptFountainBatch_vX.X.txt (e.g. v1.2)

Then select all the text in it, and copy it all to the clipboard.

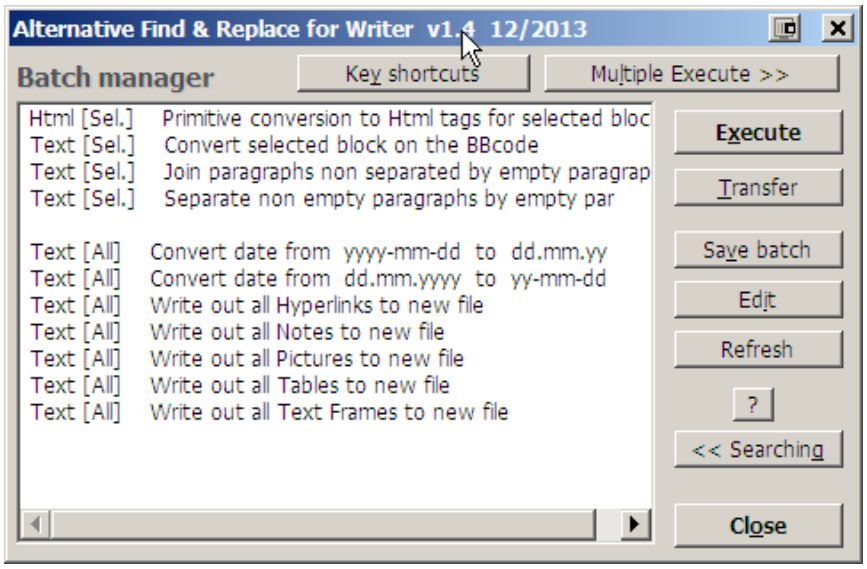
In LibreOffice, you'll see a new button on your toolbar; green binoculars:



Click that button or click the *Tools menu → Add-ons → Alternative Searching*. This will present the main dialogue box for Alt.SearchReplace:



This extension can be very handy for advanced text manipulation, by the way. You may want to play around with it. Meanwhile, find and click the button in this dialogue that says "Batch >>", which brings up the following dialog:



As you can see, it offers many built-in advanced text manipulation functions or batches. We're going to add the one which I programmed for you to these.

Find and click the "Edit" button on that dialog, which will open up the source text file that has so much handy-dandy code.

Set your cursor at the start of that text file

Press <*ENTER*> (or whatever) to create a new line, and paste (from the clipboard) all that code we copied into the clipboard in an earlier step.

Go to the *file menu → save* (to save the file).

Close the text file.

Click the bold "Close" button in the dialog we have open (that lists so many search-replace functions).

Now reopen the Alt.SearchReplace function main dialogue (again via the green binoculars button etc.).

Click the "Batch >>" button again. Hopefully the dialog that comes up will now look something like the following, which reveals new functions at the top of the list, e.g.:

Fountain to screenplay step 1

Fountain->screen step 2: delete all blank lines

Fountain->screen step 2 alt.: format blank lines

What are these new puppies for? That's what the next section explains.

Usage

For a quick demonstration, find the following file (it may be in an /examples subfolder) and open it in a text editor:

"example mangled screenplays excerpts.txt"

This document is two mangled excerpts of screenplays I've written (one not finished at this writing, arg), with deliberate ungodly syntax problems which I hope you never have to face in any screenplay.

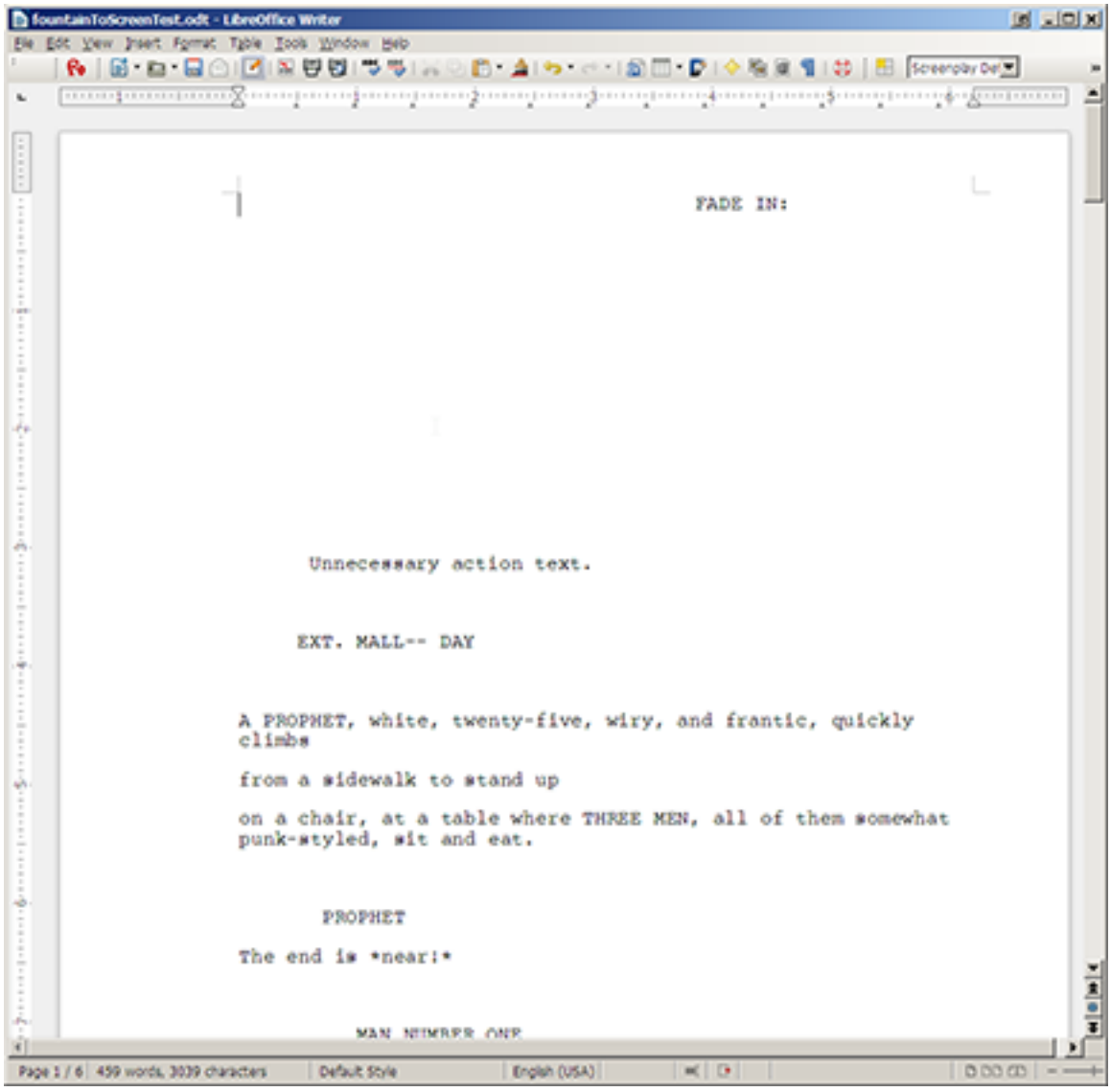
Select everything (all of the text) of that file, copy it to the clipboard (e.g. CTRL+C), and then open the following document:

"LibreOffice_ScreenplayTemplate_vx.x.x.x (standard).odt" (e.g. v1.3.4.2)

Re-save that document as e.g.:

fountainToScreenTest.odt

Select everything in that ~screenplayTemplate document (e.g. CTRL+A on Windows), and delete everything. Re-save the document under a new file name. Paste all of the example text into it (which you copied to the clipboard from an earlier step). It may look something like this:



Now open Alt.SearchReplace (the LibreOffice extension which you must have installed re the "installation" section of this document). Click the "Batch" button in the dialog to bring up this next dialog, and select the batch line that says:

"Fountain to screenplay step 1".

With that function selected (again, "Fountain to screenplay step 1"), click that bold button on the right that says "Execute."

You may see it doing its magic altering the text. First it cleans up extraneous whitespace (spaces and tabs), corrects all erroneous ellipses and dashes variants to the proper -- and ... for screenplays, changes all (V.O.) and (O.S.) variants to just (V.O.), or so my teacher taught me ;) and it also merges any multiple blank lines into one line per group. All this can be handy for technically doctoring poorly written scripts or for fixing up sources copied from HTML or pdfs.

Then, it identifies all text patterns that match supported fountain screenplay layout elements, and formats everything it finds as such (by pulling way of the styles provided with the .odt document template into which you have copied the screenplay source text). Everything it finds that doesn't match, it formats with a “Default Screenplay Style.”

Unfortunately, all of this could be faster, but it works.

(What would my ideal program be? See the “Technical notes and nits” section.)

After maybe half a minute, you'll have a properly formatted screenplay, but with extra lines left in. Why? To leave two choices open to you. If you followed this process up to this point using instead another template:

"LibreOffice_ScreenplayTemplate_vx.x.x.x (ebook).odt" (e.g. v1.3.4.2)

You would have the option to proceed in that batch list to Execute the function named:

Fountain->screen step 2 alt: format blank lines.

What will that do? To answer that requires a detour explanation:

If you're formatting for an ebook, you'll want to keep blank lines, but format them, because the only very effective way to publish screenplays as ebooks is to treat it like poetry, where a blank carriage return describes one line of vertical white space. Note also that the styles in that mentioned ebook template are adapted for ebooks; all of the vertical line space is uniformly spaced "single" (single line spacing), with indents (horizontal metrics) at 1/3 of what is standard for screenplays, to accommodate the sometimes limited horizontal screen space of ebook readers. In other words, the ebook template is a compromise between standard screenplay format and fountain, where some elements are slightly indented, but character formatting (italics etc.) are directly applied instead of being indicated by the asterisk etc. markup of fountain plain text.

End detour.

For our demonstration here, we're using the full screenplay formatting template. We will therefore, in the Alt.SearchReplace batch dialog, select the batch named:

Text [All] Fountain->screen step 2 alt.: format blank lines

Select that batch name in the list, and click the bold "**Execute**" button. In less time than before, it will

delete all blank lines. This works for our formatting purposes because the screenplay styles in this template provide the proper vertical space between lines when there are no extra carriage returns (blank lines) between screenplay elements.

Save the file again. It should look like the following document (possible in a subfolder) of this distribution:

example mangled screenplays excerpts.txt--fixed up and formatted.pdf

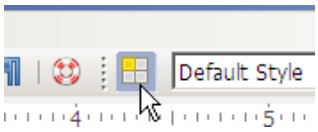
But what about a title page, and page numbering? To get those, re-open the file e.g.:

LibreOffice_ScreenplayTemplate_v1.3.4.2 (standard).odt

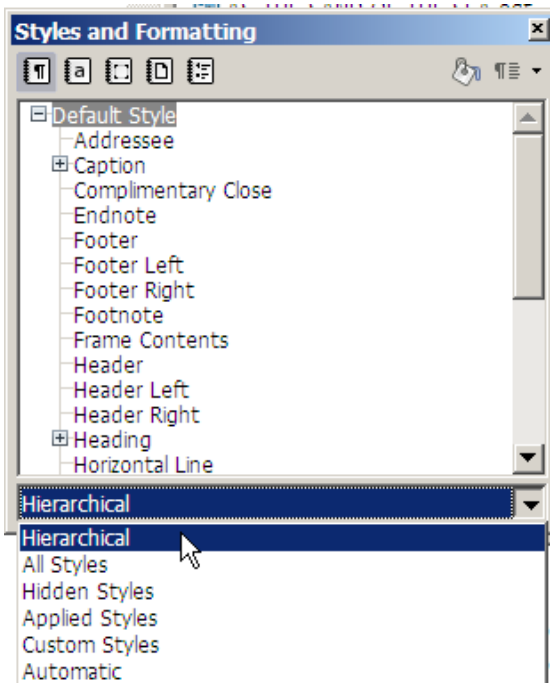
Save that document with a different name, navigate to the second page of it, delete everything on the second page, and be careful to leave the page break and page count header intact. Now go to the document we just fixed up, select everything in it, copy all that to the clipboard, come back to this document, and paste all that into this document, starting at the top of the second page. Voila! If we were formatting a "real" screenplay, now you'd only need to tweak the title page as appropriate, re-save it, and go to the *file menu → export as pdf*, and behold your new .pdf replete with proper title page and page numbering—and layout to match Movie Magic Screenwriter!

You should at least skim-read the result for formatting errors, but otherwise, that's that! If however you'd like to know more about how to control the formatting, read on.

In LibreOffice, press *F11* or click the little grid button with one yellow corner on the toolbar:



This will activate either a floating or docked panel that looks thus; click the drop-down dialog at the bottom of it and select "Hierarchical":



Now you should be able to see the screenplay styles in the document, which are part of what provides for formatting fountain to screenplay. (Incidentally, if that panel is floating and you want to dock it, try dragging it halfway off the screen until a border appears, then release it, and it will snap into a docked position.)

Using this “Styles and Formatting” panel, with the cursor on any line of the document (or really, anywhere in a block) that didn't format properly, double click any of those element names for which you'd like to format your text, and it will apply that element style to that line (and/or block) of text.

If you want to get into the guts of this template and/or tweak it for your own wants (which I urge against doing unless you have a profound need—I urge you instead to *write your screenplay* or *format it for publishing*), you can right-click any of those style names and then click *"modify..."* to look at how they work. Any changes you make to the styles in the dialog that comes up will instantaneously apply to every place in the document where the style is applied. (Tweaker-geeker's note: if you put a tick-mark in AutoUpdate under the Organizer tab, any changes you make to a styled paragraph in the the document itself will automatically propogate to the style (and therefore to all other places the style is applied). That can be handy for experimenting with and defining new styles. You usually do *not* want that, though; this is why my styles have that unticked.)

This brings us at last to:

Global screenplay font control. All of the screenplay styles inherit and apply the font specified in their parent style, "Screenplay Default Style." Right-click that style and go to *"Modify..."*

In the dialog that comes up, click on the "font" tab. If you change the font here, it will change it the entire screenplay (or, every formatted element in the screenplay, because all other elements inherit this font attribute from this their parent style).

If you therefore would like to format your entire screenplay with some other font than what the template "ships" with, this is the place to change that. I invite you to have a look at the fonts I

developed in conjunction with this toolset, distributed at:

https://sourceforge.net/projects/couriermegafont/

Also, professional (or at least more expert than I) type designers who loathe and/or want to build upon my work ;) may note the font development repository URL given at the start of this document.

You may like to install those fonts in your operating system, restart LibreOffice, and audition them for your screenplay. They might not display perfectly (or they may even display not so well) on your monitor, but in my experience and opinion, they display well as embedded in pdfs and in print. (When you export a pdf from LibreOffice, you have the option of embedding fonts—and embedding the entire .odt document, for that matter).

If you get good use out of this toolset, I'd love to hear about it; you can say hello via:

http://earthbound.io/contact

Have fun, and best of luck with your writing!

Technical notes and nits

Local styles and fountain

Regarding fountain format, you'll get incorrect local style formatting (e.g. italics/bold/underline) if you incorrectly combine different formats. For example, if you have an entire block which is bold, and in the middle of it you want something bold italic, you must format it like the following;

Correct usage:

```
**Bold text** ***bold italic text*** **bold text**
```

Incorrect usage:

~~**text *which will not format the way you intend* text**~~

These problems may be a limitation of how my batch parses formatting marks. I have no plans to fix that if so. Lastly, this toolset supports pagebreaks, which in fountain are three equal signs in a row, on one line:

===

I left that coded in the batch even though I don't use it (my template has a title page and page break already, and that's the only place I'll use a page break).

Screenwriting syntactical nits

The formatting batch will interpret any text in parenthesis alone on one line as a dialogue

parenthetical. It's hard for me to imagine where that wouldn't be a parenthetical in a screenplay, but still you should know.

Also, please be advised that in your screenplay, if any ALL CAPS text that isn't a scene slug ends up alone on one line, the formatting batch is likely to interpret that as a character slug, and format it as such—probably with any immediately subsequent action incorrectly formatted as dialogue. That's more likely to happen if you are one of those writers with a ridiculous comic-book-style writing habit of liberally peppering your script with meaningless ALL CAPS all over the action—which is in my opinion a poor and too common idea. If that is the case, and you don't watch it, you or the folks who buy your screenplay could import your screenplay to Movie Magic Screenwriter, and run a character breakdown, and then ask: who is this ONCOMING CAR character? And why does he/she have a gerond in his/her name? Well, he/she isn't a character—so why write it that way? Please, no. In my opinion, all caps should be very sparing, and inline with the action.

Regarding the syntax fixup which the formatting batch does as part of formatting:

For dashes:

```
text --text, text - text, text—text
```

Such incorrect dashes (note the em-dash) are corrected to e.g.:

```
text--text
```

Or double-dash, no spaces, no em-dash.

For ellipses, variations such as:

```
text . . .text, text ..text text…
```

(Note the ellipses glyph) etc. are corrected to:

```
text...text
```

Or they are correcte to three dots, with no spaces.

All irregular dubbing extensions such as:

```
(O.S.)
```

```
(V. O.)
```

Etc., even if they have wonky spaces like that latter, are changed to just:

(V.O.)

With no spaces. Why? Both mean the same thing, but V.O. is more practical, and why use two different slug appendages that mean the same thing? For that matter, (MORE) and (CONT'D) tags are unnecessary. The facts they convey are already very obvious from context, so that the only (impractical) effect those tags have are to clutter a page.

Alt.SearchReplace general technical information

The Alt.SearchReplace extension which this toolset relies on uses regular expressions (and provides a framework to use a series of regular expressions in a batch to process text). However, owing to the technical complications of how those expressions can interact with Open Document format as they do, they can be very difficult. In other words, rigging this toolset to use those regular expressions was severely difficult. But I did it!

Ideal tool?

To re-do this, first of all I wouldn't, and I'd pay a programmer to do so if possible—but I'd make it a console application. It would be constructed from the ground up to patch together the requisite bits of a properly formatted screenplay .odt, using LibreOffice's open document creation API, and details of the styles I made for this templay, and using the regexes available in the fountain developer's toolkit.