

nprstuff

Tanim Islam

November 23, 2015

I like NPR, so I made some scripts to download my favorite programs from NPR. For now, I have something that downloads [Fresh Air](#), [Wait Wait..Don't Tell Me](#), and [This American Life](#). This package can probably, straightforwardly be extended to other NPR and PRI programs.

Although this project started off as a way to download these three programs, I have expanded it to include a grab bag of altogether different types of functionalities. What remains the same? This distribution consists mainly of executable python scripts.

I organize this document into the following sections: [Core Functionality](#), [New Functionality](#), [Graphics Functionality](#) (in folders `gui` and `gui2`), and a small section called [Oldstuff](#).

This document was converted from a \LaTeX source using [Pandoc](#), via

```
pandoc -s README.tex -o README.md
```

Core Functionality

This consists of functionality to grab episodes from [Fresh Air](#), [Wait Wait..Don't Tell Me](#), and [This American Life](#). These consist of the following pieces of python code:

- `npr_utils.py` contains common utilities to get the proper metadata for NPR programs, to name these media files in the proper date format, and to get the full paths to the [LibAV/FFMPEG](#) and [HandBrakeCLI](#) tools to create the NPR programs in m4a and mp3 formats (among other functionalities).
- These four files handle NPR Fresh Air downloads: `freshair.py`, `freshair_crontab.py`, `freshair_fix_crontab.py`, and `freshair_by_year.py`.

- `freshair.py` is the main executable that downloads NPR Fresh Air episodes, converts them to m4a format, and then applies correct meta-data. The help screen for this command line tool is here,

Usage: `freshair.py` [options]

Options:

```
-h, --help            show this help message and exit
--dirname=DIRNAME     Name of the directory to store the file. Default is
                        /mnt/media/freshair.
--date=DATE           The date, in the form of "January 1, 2014." The default
                        is today's date, November 14, 2015.
--debug              If chosen, run freshair.py in debug mode. Useful for
                        debugging :)
```

- `freshair_crontab.py` downloads an NPR Fresh Air episode on a given weekday. It should be called by a cron job that should be run every weekday.
 - `freshair_fix_crontab.py` tries to re-download NPR Fresh Air episodes that may be incomplete – defined as shorter than 30 minutes – and which are 90 days or older. This executable searches through the library of all NPR Fresh Air episodes, and tries to re-download older, possibly incomplete episodes.
 - `freshair_by_year.py` downloads all the NPR Fresh Air episodes in a given year.
- These four files handle NPR Wait Wait downloads: `waitwait.py`, `waitwait_realmedia.py`, `waitwait_crontab.py`, and `waitwait_by_year.py`.

- `freshair.py` is the main executable that downloads NPR Wait Wait episodes, converts them to m4a format, and then applies correct meta-data. `waitwait_realmedia.py` is a python module that allows one to download NPR Wait Wait episodes older than 2004, which are in [RealMedia](#) format. The help screen for this command line tool is here,

Usage: `waitwait.py` [options]

Options:

```
-h, --help            show this help message and exit
--dirname=DIRNAME     Name of the directory to store the file. Default is
                        /mnt/media/waitwait.
--date=DATE           The date, in the form of "January 1, 2014." The default
                        is last Saturday, November 14, 2015.
--debugonly           If chosen, download the NPR XML data sheet for this Wait
                        Wait episode.
```

- `waitwait_crontab.py` downloads an NPR Wait Wait episode on a given Saturday. It should be called by a cron job that should be run every Saturday.
- `waitwait_by_year.py` downloads all the NPR Wait Wait episodes in a given year.
- `thisamericanlife.py` *manually* downloads a given episode number of This American Life. This executable uses a custom online archive for older This American Life episodes that are described [here](#). The help screen for this command line tool is here,

Usage: `thisamericanlife.py` [options]

Options:

```
-h, --help            show this help message and exit
--episode=EPISODE     Episode number of This American Life to download.
                        Default is 150.
--directory=DIRECTORY Directory into which to download This American Life
                        episodes. Default is /mnt/media/thisamericanlife.
--extra=EXTRASTUFF    If defined, some extra stuff in the URL to get a This
                        American Life episode.
```

New Functionality

This consists of newer functionality that does not download NPR episodes, nor can one straightforwardly modify them to download NPR episodes. These consist of the following pieces of python code.

- `autoCropImage.py` automatically crops image (png, jpeg, tiff, etc.) files to remove whitespace. The default whitespace color is `white`. The help screen for this command line tool is here,

Usage: `autoCropImage.py` [options]

Options:

```
-h, --help            show this help message and exit
--input=INPUT         Name of the input file.
--output=OUTPUT       Name of the output file. Optional.
--color=COLOR         Name of the color over which to autocrop. Default is white.
```

- `convertImage.py` uses the [CloudConvert REST API](#) to *smoothly and without pain points* convert and resize SVG images to PNG images of the same base name. The help screen for this command line tool is here,

Usage: convertImage.py [options]

Options:

-h, --help show this help message and exit
--filename=FILENAME Name of the input SVG file.
--width=WIDTH If defined, new width of the file. Optional

- `changedates.py` changes the creation date of JPG and MOV files, that my Canon digital camera creates, by up and down one year. I created this tool because my Canon digital camera does not set the right year on the creation date for image files it creates. This caused problems when I uploaded those images to [Google Picasa](#) or [Google+](#). The help screen for this command line tool is here,

Usage: changedates.py [options]

Options:

-h, --help show this help message and exit
--dirname=DIRNAME Name of the directory to look for jpeg files.
--movs If chosen, process MOV files instead.
--minus If chosen, subtract a year from the files.

- `music_to_m4a.py` can convert a single file from mp3/ogg/flac format to m4a format while preserving music file metadata, and can optionally set the total number of album tracks and the album cover if the music files is in an album. It can also rename an m4a music file into the format "*artist name - song name.m4a*." The help screen for this command line tool is here,

Usage: music_to_m4a.py [options]

Options:

-h, --help show this help message and exit
--inputfile=INPUTFILE Name of the input audio file to convert.
--outfile=OUTFILE Optional name of the output file.
--tottracks=TOTTRACKS Optional total number of tracks in album of which song is a part.
--albumloc=ALBUMLOC Optional path to location of the album cover image file. Must be in JPEG or PNG.
--quiet If chosen, then verbosely print output of processing.
--rename If chosen, simply rename the m4a file to the form <artist>.<song title>.m4a

- `download_surahs.py` downloads recorded surahs ([Abdur-Rashid Sufi](#)) to a directory of your choice. The help screen for this command line tool is here,

Usage: download_surahs.py [options]

Options:

-h, --help show this help message and exit
--outdir=OUTDIR Directory to put this data into. Default is
/mnt/software/sources/pythonics/nprstuff.

Graphics Functionality

This section describes the two graphical tools I have developed: `gui` matches a small subset of functionality that the [Readability](#) tool handles excellently; `gui2` is a [PyQt4](#) GUI front-end to the [Readability](#) API.

GUI: Media Website Text Formatter

This GUI can read from the following media websites: [Lightspeed Magazine](#), [Medium](#), [The New Yorker](#), [The New York Times](#), and the [Virginia Quarterly Review](#). Here is a screenshot! The screenshots of the save file dialog and the

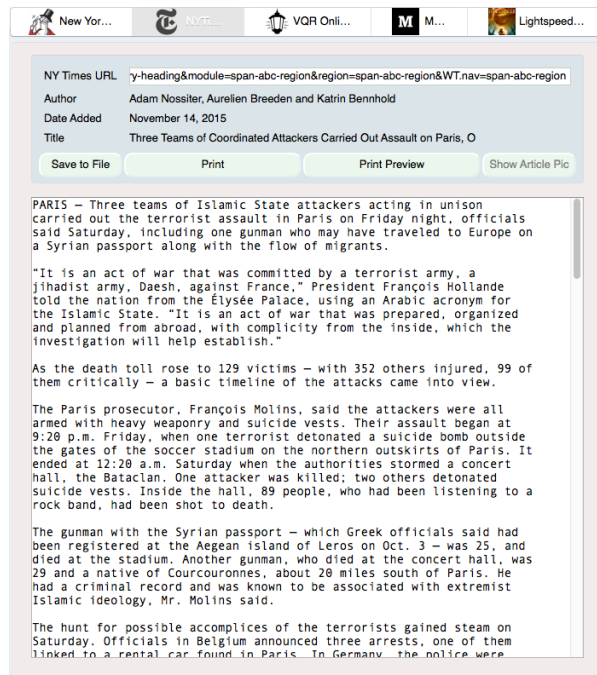


Figure 1: A screenshot of the GUI reader, converting the URL for the [The New York Times](#) into text. Note the separate icons above for the five media websites from which this GUI can read.

print preview dialog are shown [Fig. 2](#) and [Fig. 3](#), respectively. Note, here I do not support or maintain this tool after I found out about [Readability](#).

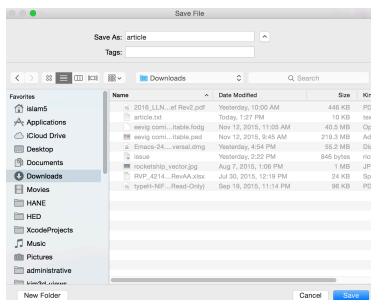


Figure 2: The GUI screenshot of the save dialog.

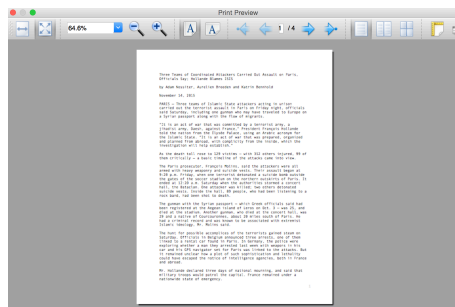


Figure 3: The GUI screenshot of the print preview dialog.

GUI2: Readability GUI Front-End

This is the PyQt4 GUI front-end to [Readability](#). A screenshot of the list of articles widget is shown in Fig. (4), and a screenshot of the article text widget is shown in Fig. (5). A screenshot of the font changing dialog, the **Font** button,

title	date	time	word count
Aziz Ansari on Acting, Race and Hollywood	11/14/2015	20:49:51	1630
America's Poorest White Town: Abandoned by Coal, Sw...	11/13/2015	00:38:14	7236
The Revolutionary	11/12/2015	17:15:23	12817
Tricked and Indebted on Land, Abused or Abandoned at ...	11/10/2015	01:58:56	3829
"Honey Bunny"	11/07/2015	06:16:00	5645
The Displaced: Hana	11/06/2015	22:22:11	4991
Can the G.O.P. Deal With Inequality?	11/06/2015	14:56:50	7096
A College Romance That Led to Murder	11/04/2015	20:19:06	11629
In Heroin Crisis, White Families Seek Gentler War on Dr...	10/30/2015	22:41:49	2678
Can Fast Food Get Healthy?	10/30/2015	05:38:14	6646
Hey, You Got Something to Eat?	10/29/2015	05:29:41	841
Syrian Refugees in Jersey City Are Among Few to Start...	10/24/2015	16:22:59	1783
A Global Chill in Communist Demand like America's Ma...	10/14/2015	00:13:00	1000

Figure 4: List of bookmarked (stored) articles under the user's account.

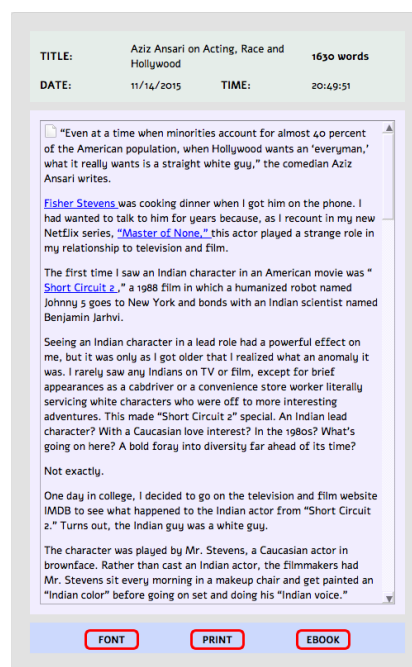


Figure 5: The text form of the article's content, with working dialogs for **Font** and **Print Preview**.

is shown in Fig. (6). A screenshot of the print preview dialog, the **Print** button, is shown in Fig. (7). In the immediate future, I plan on at least implementing

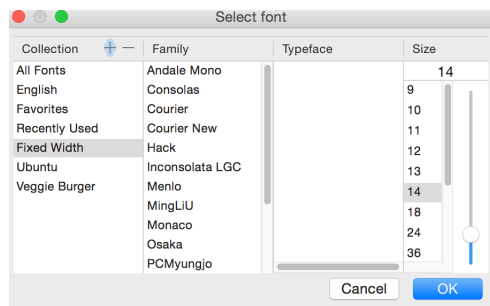


Figure 6: The font changing dialog launched by the **Font** button in the article text widget. The initial preference for the font to display are fixed width fonts.

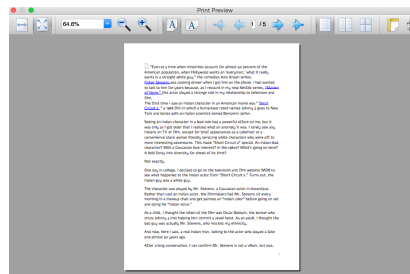


Figure 7: The print preview dialog launched by the **Print** button in the article text widget.

the following, all using the Readability API.

- EPUB button, to create the article in [EPUB](#) format.
- Adding and deleting articles through the article list widget.

Oldstuff

These are tools that I do not maintain, located in the `oldstuff` folder, but which others may find useful. These are pieces of code that I have started, but which are unmaintained. These are the following pieces of code: `freshair.sh`, `waitwait.sh`, and `google_pull_contacts.py`.