# Steps in Optical Character Recognition with Tesseract

Joel Kalvesmaki

License: CC BY-4.0

The items provided on this handout are intended to accompany the first of four 2021 North American Patristics Society workshops, devoted to optical character recognition. We will walk through the steps of OCR work with Tesseract, based on two examples:

John Bunce’s 1759 English translation of John Chrysostom’s *On the Priesthood.* <https://archive.org/details/ofpriesthoodinsi00john>

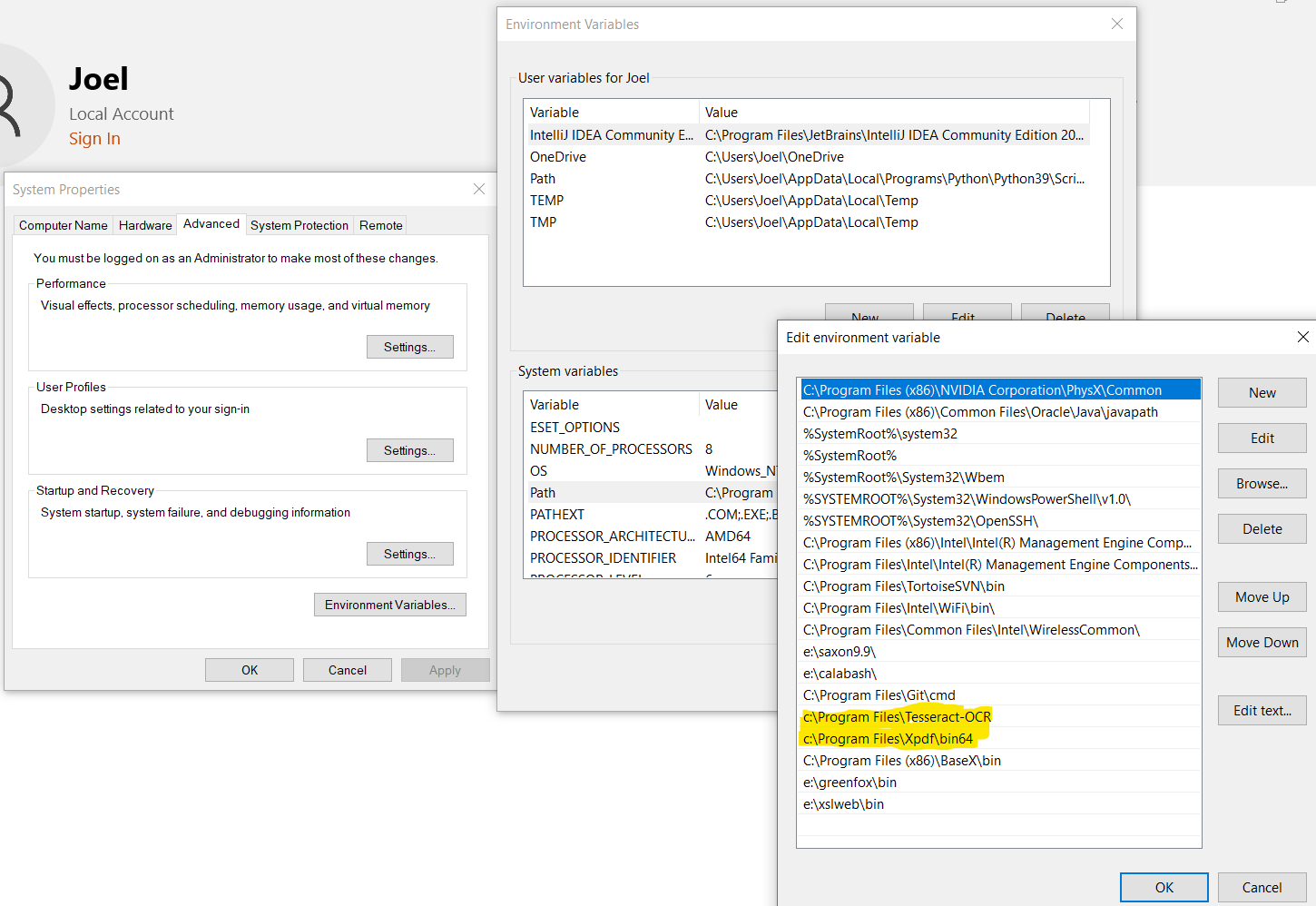
J. A. Nairn’s 1906 Greek edition of John Chrysostom’s *On the Priesthood*. <https://archive.org/details/DeSacerdotioOfStJohnChrysostom>

# Installation of Software

Download and install tesseract: <https://tesseract-ocr.github.io/tessdoc/Downloads.html>

Optional: download and install XpdfReader (for Pdftotext and Pdftoppm): <https://www.xpdfreader.com/download.html>

Windows users might need to adjust their system. Windows key > “environment variables” > System variables > Path > Edit > add paths for Tesseract and Xpdf (if not present):



Get a plain text processor:

Windows: Notepad, or download Atom (<https://atom.io/>), Notepad++ (<https://notepad-plus-plus.org>), Sublime Text (<https://www.sublimetext.com/>).

Windows: TextEdit, or download Atom (<https://atom.io/>), BBEdit (<https://www.barebones.com/>), Sublime Text (<https://www.sublimetext.com/>).

## Set up files

Download PDFs/images and set them up as you like in the directory of your choice.

## Use the shell

**Open the shell**. Windows: cmd. Mac: Applications → Utilities → Terminal

**Navigate to your files**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mac / Linux | Windows | Purpose | Examples |  |
| cd | cd | change directory | cd ..  cd files | go up one level  go into subdirectory files |
| ls | dir | list directory contents |  |  |
| ↑ | ↑ | show previous command |  |  |
| ↓ | ↓ | show next command |  |  |
| [tab] | [tab] | autocomplete |  |  |

You can find lots of other commands by searching online.

## Try a non-OCR approach

You can try to scrape the text out of the pdf, if it exists:

pdftotext ofpriesthoodinsi00john.pdf bunce.txt

That line will scrape all the text from the local pdf and put it into a file called bunce.txt.

There are a lot of other options with pdftotext. To see them all, just type pdftotext.

## Convert your PDF to images

Tesseract handles jpegs and pngs, but it does NOT handle TIFFs, nor does it take PDFs. You can convert the images to PNGs as follows:

pdftopng -r 300 -f 37 -l 40 ofpriesthoodinsi00john.pdf bunce300

That line will look for pages 37-40 of the local pdf, and save each page at 300 dot-per-inch resolution, saving it incrementally to bunce300-000037.png, bunce300-000038.png, etc.

There are a lot of other options with pdftopng. To see them all, just type pdftopng.

## Convert your images into text

Now the fun begins! All you need to do is invoke tesseract as follows:

tesseract bunce300-000037.png bunce-037 -l eng

That line asks Tesseract to process the selected image and place the output at a file called bunce-037.txt, and to treat the input as English.

If you are working with another language, be sure to use the 3-letter name of the language, e.g., grc for Greek, lat for Latin, syr for Syriac, cop for Coptic. If your text has a mix of languages, just join the language codes together with a plus sign, putting the most important language first. For example, -l grc+eng could be good for a critical edition where there are Latin sigla to tease out.

There are some other options tesseract offers. To see them all, just type tesseract, and visit the project page, <https://tesseract-ocr.github.io/tessdoc/Home.html>.

## Check your results, calibrate

Once you’ve tried a page, check your results. Are they what you expected? Can you improve the image quality? Try some tips here:

<https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html>

Don’t do a lot until you’re happy with what you have.

It is a good idea to create one or two pages that have the text as expected, then to compare your results against those versions. There are a variety of file or string comparison tools out there. A very cheap way to compare two texts is to make two Word or LibreOffice files, then choose the file comparison tool. It will show you where the differences are, and you can eyeball them.

## Set up a pipeline

In many OCR projects, one wants to handle hundreds or thousands of pages. Tesseract does not have a built-in method of automation, but you can easily build one yourself.

Look at the spreadsheet **ocr batch file maker.xlsx** (under subdirectory #6). You can easily change the values as you need, then copy the last column into the batch file called tessalot.bat. This is called a batch file, and it is used in Windows machines. Now put that file into the local directory where your target files are, then go back into the shell and navigate to the directory. Now just type tessalot. This works great for Windows, but you can do something similar in Mac, via a file called a shell script, with an extension .sh. And it’s pretty much the same kind of routine, but when you navigate to the directory, you want to type ./tessalot.sh.

## Bringing it all together

So at the end you’ve got perhaps hundreds of files that you need to put into a single file. In Windows, at the prompt, try this:

type bunce\*.txt >> bunce-all.txt

That will concatenate (join) all the files that start with “bunce” and end with a “txt” extension into a giant file called bunce-all.txt. There’s a similar command you can do in Mac or Linux:

cat bunce\*.txt >> bunce-all.txt

Once you have all the text in a single file, you can then start doing cleanup, by making global changes to the text. Check out our second workshop with Jamey Walters, on regular expressions for some power tips for cleaning a file.