MIDDLESEX Community College

Tools and Technologies for Tech Writers 2022

# Tips and Tricks

## Notices

This document was prepared as a handout for the Middlesex Community College Tools and Technologies for Technical Writers class, Winter semester 2022.

Prepared by Zoë Lawson, course instructor.

# Contents

# Introduction

Over twenty years of techwriting, I've learned a thing or two that doesn't always fit into nice, descrete categories.

# File Explorer configuration

Default settings for Windows File Explorer are terrible. I have customizations I make every single time I set up a system.

Both Windows and Mac systems are trying to be "helpful" and make it "easy" for you to use a computer. They're trying to make computers "less scary".

Unfortunately, sometimes this gets in the way of using them.

1. Open Windows File Explorer.
2. Open the **View** ribbon.

   If the View option isn't visible, press the **Alt** key and the menus should appear.
3. Select **Navigation Pane** and confirm **Navigation Pane** is selected.

   This should make sure you always get the tree view. When I'm moving between folders, this really helps me.
4. As long as I'm not in a folder full of images, I select **Details** for the view.
5. Optional: If I'm using a source control tool that keeps locked files read-only, I add the attributes to the view.
   a. Right-click the header row (where it says Name, Date modified, Type, and Size) and select **More**.
   b. Select **Attributes**.

   Now, if a file is read-only, I can see it's marked with `R`.
6. In the View ribbon, select **Options** > **Change folder and search options**.
7. Go to the **View** tab and change the following:
   - Select **Always show menus**.
   - Select **Disply the full path in the title bar**.
   - Select **Show hidden files, folders, and drives**.
   - Clear **Hide extensions for known file types**
   - Under Navigation Pane, select **Expand to open folder**.
   - Under Navigation Pane, select **Show all folders**.
8. Click **Apply to Folders**.
9. If the `Do you want all folders of this type to match this folder's view settings` click **Yes**.
10. Click **OK**.

As a tech writer, you're going to care about what types of files you have. Whether that image is a `.jpg` or `.png` is really important. You're going to end up with files named the same but with different extensions. You want to be able to see these easily.

# Paths

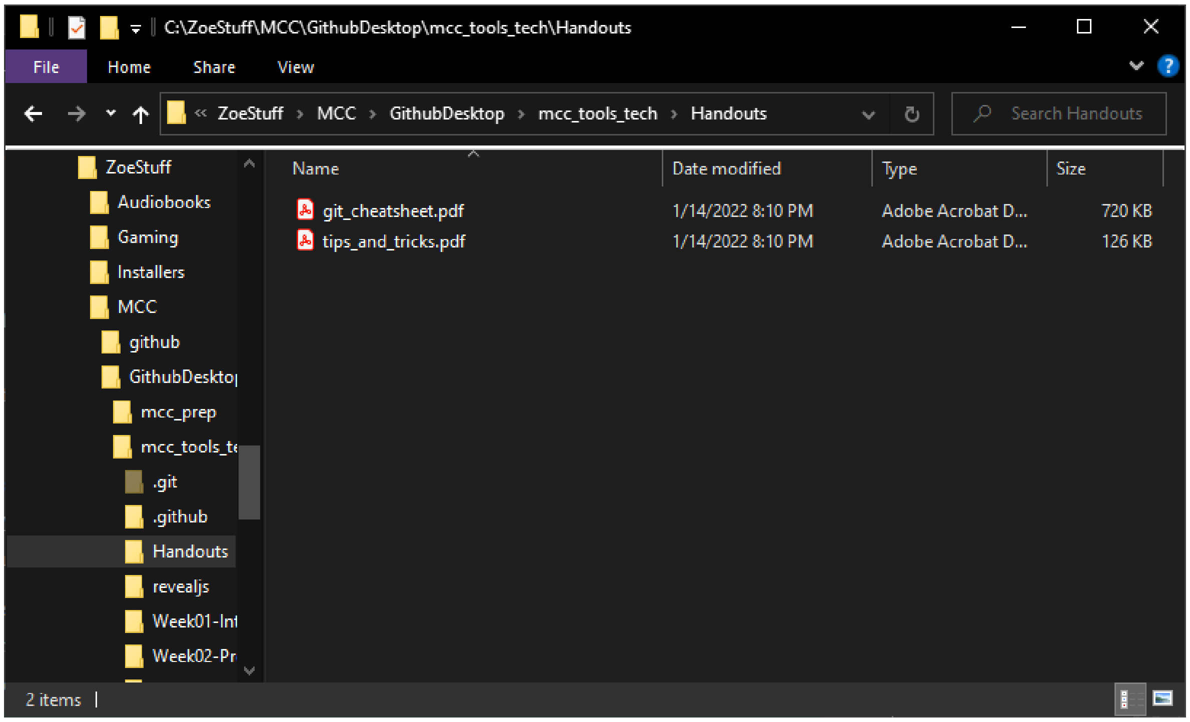
When working with computers, you often have to "write directions" on where to find a file. These are called 'paths'.

Computers have tons of information in them. This data and the tools to view and use the data is organized for humans into files and containers usually called folders or directories. The actual bits and bytes may be scattered all over a harddrive, but logically, computers present this data to us as files and folders. Files can be in folders, but folders cannot be in files. Folders can be nested inside of other folders.

Humans need to be able to find files and applications, so we need some sort of addressing system. Computers use bits and bytes and pointers n' things that are very hard to remember, unless you like remembering hex codes. So, someone figured out that you can write out the "path" to the file using folder names.

In the Unix world, someone decided to use forward slashes (/) as the separator between folder and file names. To avoid copyright infringement (or to be annoying) DOS (i.e. Windows) uses back slashes (\). This has caused all sorts of confusion and annoyance and bugs in the computer world ever since.

Most computer languages like to use forward slashes. Notice URLs, they use forward slashes. Notice all the paths in GitHub, they use forward slashes. Often, when linking to files for cross-references or images, you use forward slashes.



| Name | Date modified | Type | Size |
|---|---|---|---|
| git_cheatsheet.pdf | 1/14/2022 8:10 PM | Adobe Acrobat D... | 720 KB |
| tips_and_tricks.pdf | 1/14/2022 8:10 PM | Adobe Acrobat D... | 126 KB |

This is a screenshot of my GitHub Desktop folder location for this handout. You can see the path in the title bar: `C:\ZoeStuff\MCC\GithubDesktop\mcc_tools_tech\Handouts`. Since this is a Windows system, it's using backslashes. In the Navigation Pane/Tree View, you can see all the folders and how they nest.

If I want to know the precise location of this handout on my computer, it is `C:\ZoeStuff\MCC\GithubDesktop\mcc_tools_tech\Handouts\tips_and_tricks.pdf`. This is called a *fully qualified path*. This tells me exactly where the file is, on my computer.

This is great...as long as I am working on my computer. If you want to find the file on your computer, it's not quite as useful.

Sometime people use *variables* in path descriptions. Common formatting conventions include italic text or surrounding with angle brackets. So I might write about this location as `<`*`GitHub Desktop Repo Location`*`>/mcc_tools_tech/Handouts/tips_and_tricks.pdf`. The string `<`*`GitHub Desktop Repo Location`*`>` indicates whatever directory I picked for cloning the repository. That could be the default of `C:/Users/`*`User Name`*`/Documents/GitHub` or the path I picked, `C:\ZoeStuff\MCC\GitHubDesktop`.

If you pick a formatting convention for variables, make sure you include some non-formatting related indicator. If you just make variables italic, a screen reader may not recognize the font change. If you include something not format related, such as adding the angle brackets (< >), it makes the difference more accessible.

This format of writing paths is extremely common in technical documentation. The only difficulty is standardizing how you refer to your variables so you don't have a mishmash of *Install_dir*, *Product Installation Directory*, and *Installation_Dir*.

While using string variables is great for tech doc, it's not super useful for computers. Yes, there are variables in programming languages, and you absolultely use them in paths in programming, but generally speaking, writing tech doc isn't programming. So there's another type of path you use all the time, a *relative path*.

A relative path is directions to a file based on where you're starting.

- `filename.txt` - the file is in the same folder as where you are starting.
- `directory/filename.txt` - the file is in the directory folder. The directory folder is in the same folder where you are starting.
- `../differentDirectory/filename.txt` - the file is in the differentDirectory folder, which is in the "parent" folder to where you are starting. With a .., you go "up" a folder.

For example, I'm starting in the Handouts folder and I want to get to the Week 2 homework folder. The Week 2 folder is not in the Handouts folder, so I go up a folder.

```
../
```

This puts me into the `mcc_tools_tech` folder. Goody! The Week 2 folder is in this directory, and I want to go into it.

```
../Week02-ProgressiveInfo
```

Now that I'm here in the Week 2 folder, there's the homework folder I want. So the relative path from the `Handouts` folder to the Week 2 Homework folder is:

```
../Week02-ProgressiveInfo/Homework
```

This relative path only works from certain locations, such as from the Handouts directory. However, this path will always work from the Handouts directory to the Week 2 Homework folder in the mcc_tools_tech repository, no matter what computer I'm on, or where you clone the repository.

Learning how to make relative paths between files is really useful. You use them all the time for cross-references and links to images.

# Regular Expressions

Regular expressions (regex) are a way of augmeting searches to be more specific.

Regular expressions are a common "language" used to search for text strings. Many different scripting and programming languages use them. Some find and search and replace tools also allow them. (Notepad++ is a text file tool that uses them.) You can even sometimes do some search and replace.

There are several different flavors of regex. But there many options that are relatively standard. Here's a few common expressions:

Regex are case-sensitive. Searching for `Z` is different than searching for `z`.

| | |
|---|---|
| **.** | Any single character |
| **[]** | A set or range of characters. `[xyz]` finds the character x, y, or z. `[a-z]` finds any lowercase letter between a and z. You can also search for numbers, such as `[0-9]`. A common expression is [a-zA-Z] which finds any upper case or lower case letter. [a-zA-Z0-9] finds any alpha-numeric character. |
| ***** | Zero or more times. In general `.*` finds most anything. |
| **\** | Escape character. `.` finds most anything. `\.` finds a period. If you need to find a backslash, you can double it, `\\`. |
| **^** | When used inside square brackets, not whatever. `[^a-z]` finds any character that isn't a lowercase letter. |

There are many, many other options, but you can find your favorite cheat sheet, or the flavor you need for whatever application you're using on the internet.

Some sites to get you started:

- https://www.rexegg.com/regex-quickstart.html
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions/Cheatsheet
- https://cheatography.com/davechild/cheat-sheets/regular-expressions/

# Open source tools

There are a ton of great free tools available for you to use.

Always, check with your employer to figure out what their policy is for third-party open source tools. You may not be allowed to use them, or only tools under specific licenses (GNU vs Apache, for example), or they may have a special place to download 'blessed' versions.

These are the tools I'm aware of. There may be newer ones or better ones available, but these should get you started.

## 7zip

A zipping tool with a command line.

Once upon a time, Windows did not come with the **Send to compressed folder** option. There was WinZip, but you had to pay for it.

7zip is a free tool that makes zip files. Windows now has a built-in utility, but it doesn't have a lot of options. If you install 7zip, you can make zip files and also update contents of zip files. 7zip includes an integration with File Explorer, so it's easy to zip up or extract files.

The big plus is that it comes with a command line. If you want to get into scripting, being able to automatically zip up files is extremely useful. Most real scripting languages such as Python include zipping libraries, but Windows scripting options didn't. (Unix generally comes with `zip` or `gzip` commands.)

### Reputable download

https://www.7-zip.org/

## Audacity

If you get into making videos with an audio track, you may want some additional audio editing tools beyond what comes with your video tools. Audacity is a decent free audio editing tool.

If you need to edit audio, Audacity is great. I was introduced to it when I got a USB turn table and I was converting record albumns to MP3s. I could use Audacity to divide up the record into tracks.

If you're doing video with a voice over, you can use Audacity to clean up your audio clips.

### Reputable download

https://www.audacityteam.org/

## Gimp

If you want a free image editing application, this one has a bunch of features.

Photoshop is one of the standards, but Adobe products are not cheap. I have not used gimp much because I have spent money on other tools, but many folks I know have used it and are very happy with it.

### Reputable download

https://www.gimp.org/

## Imagemagick

This is a command line tool, so may not be your cup of tea, but if you have to do bulk conversion, it may be worth looking into.

This is a very, very powerful tool. It can work with many different types of files and convert them to other types. It can do all sorts of resizing, resampling, etc.

I've used this tool to automate resizing images. If you want to have different sized images with different resolutions for PDF vs online, you can use this tool to automatically take one image and create the different versions you need, or for making thumbnails. It's also great for conversion work if you have to update a whole bunch of JPG files to PNG files.

### Reputable download

https://imagemagick.org/index.php

## Inkscape

Resizing images is pain, so having a good vector image editor is useful.

There are two major types of images, bitmaps and vector diagrams. Oversimplifed, bitmaps images are files that save the information as color information and pixel position. Vector diagrams are image files that save the information as shapes and math with properties. In general, photos are bitmaps. Illustrations are probably vector diagrams. Photoshop and CorelPhoto edit bitmaps. Adobe Illustrator and Corel Draw are vector editing applications.

Inkscape is an open source vector drawing program. You can save them as SVG files. Today, most browsers can render SVG images. Because SVG images are vector based, you can zoom into them and they will always look good.

### Reputable download

https://inkscape.org/

## kdiff3

At some point in time, you will be working text files. Your going to need to compare text files to figure out what's going on.

There are many different comparison tools out there. This is one I found years and years ago that I have been very happy with.

### Reputable download

Source: https://invent.kde.org/sdk/kdiff3

If you go to the README, you can find the link to the download.

https://download.kde.org/stable/kdiff3/?C=M;O=D

## Notepad++

You have to edit text files. Or tidy up stuff that's easier to work with as text. Or change a file's encoding. Notepad++ is a fantastic text editor.

You are going to need a text editor. Sometimes you will be working with text files, such as properties files or raw HTML. Other times you will be cutting and pasting between applications that add extra bits. (Have you ever copied something from a web page and pasted it into a Word file and got extra junk?) Yes, all computers have some sort of text editor, but plain old Notepad doesn't have extra features. Having a good text editor is invaluable.

Notepad++ is my favorite text editor. It's familiar with a whole bunch of common coding languages and provides highlighting. You can change file encoding (e.g. English vs Russian) and also change line endings. You can search and replace across files. You can perform searches using regular expressions.

### Reputable download

https://notepad-plus-plus.org/

## Pencil

This tool was introduced to me as a UI mockup tool. You can also use it to make basic diagrams.

Being able to throw together a UI design is sometimes part of your job. Being able to put together basic workflow diagrams is a common requirement for techdocs. This is a nice tool to be aware of and can be worth using.

### Reputable download

https://pencil.evolus.vn/