# Separating the data from the CHAFF

*By Matthew Murray 🦇 (CU Boulder), Fernando Rios (University of Arizona), and Seth Erickson (UCSB).*

When you've used computers long enough you've probably come across some files or folders (henceforth called directories) that don't look quite right. The names are similar to, though not exactly the same as, other files and directories you have, but they don't follow the standard naming conventions. Additionally, if you open one of the files, they don't contain any information that seems useful or relevant. You didn't make these files and you don't want them, so what are they and where did they come from?

These files are CHAFF: Concealed, Hidden, And Forgotten Files. Exactly what these files and directories are and what they do depends on several factors, but they are generally used by the programs that created them as caches, temporary backups of files being edited, or operating system-specific metadata. These files are often hidden unless a user explicitly looks for them or shares data with someone running a different operating system. Thankfully, once they've been found, they can usually[1] be deleted safely!

This post will start by looking at files created on MacOS and Windows separately as, while there are similarities between them, they serve different purposes. We end with some considerations when curating a dataset for archiving and provide a few potentially useful commands to help remove CHAFF or keep them from getting into an archive in the first place.

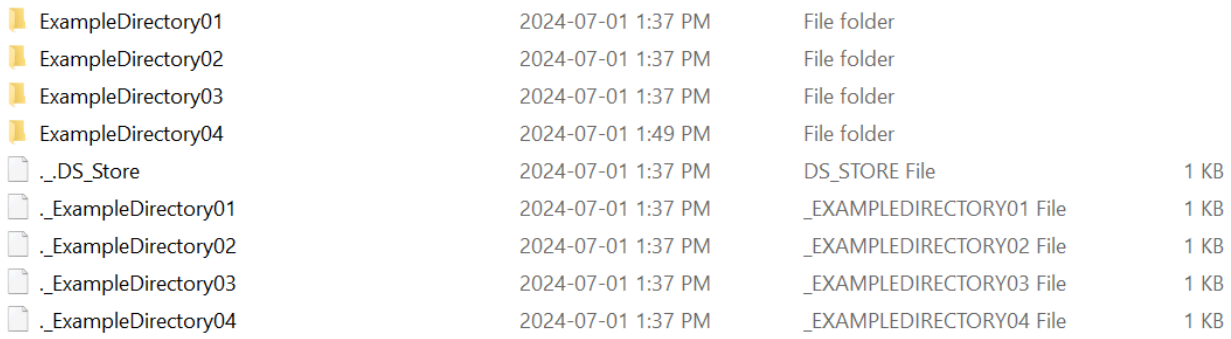
## MacOS

If you use MacOS you probably haven't encountered these yourself, but if you've ever sent or uploaded a .zip file[2] to a colleague using Windows, they've almost certainly seen the following:

- Files named .DS_Store
- Filenames starting with ._
- Directories named __MACOSX

[1] There is always the possibility that what looks like an irrelevant file turns out to be a crucial part of a dataset. If you encounter any undocumented files when curating a dataset, the first step should always be to contact the authors and ask them for more information.

[2] Throughout this document we use the term ".zip file" to represent all archive file formats. Other archive formats include .tar, .rar, .7z, and .gz. For more examples, see the Wikipedia article on "List of archive formats."

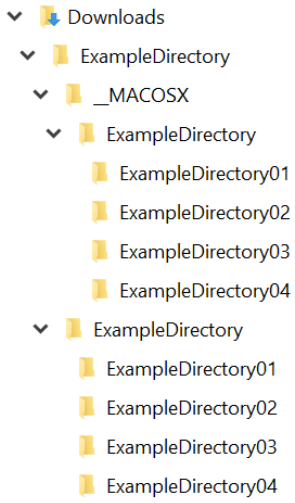



*[Figure 1: A screenshot of Windows' File Explorer showing directories titled ExampleDirectory01 to ExampleDirectory04. There are also files titled ._DS_STORE and ._ExampleDirectory01 to ._ExampleDirectory04. Each of the ._ files is 1 KB in size.]*

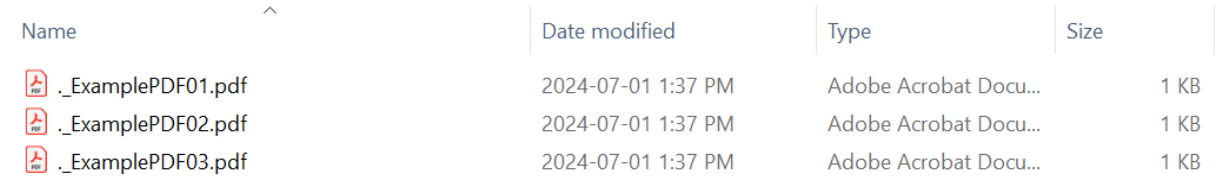
These are files that Apple's Finder application (equivalent to Windows' File Explorer) creates automatically for every file and directory that it interacts with.

- .DS_Store
    - DS_Store stands for "Desktop Services Store" and these files are used to keep track of metadata used by Finder–such as display options or icon positions–and Spotlight's search features.
- ._
    - These are "extended attributes" files and directories which contain metadata that is not embedded into the files themselves. This can include things such as "tags" created by Finder.
- __MACOSX
    - These directories contain an identically named directory structure containing ._ files.

If you use Apple products then you probably haven't seen these files because MacOS (as well as many Unix and Linux distributions) treat files beginning with a "." character as invisible and the files are therefore concealed or hidden. If you create an archive (such as a .zip file) of a directory while using MacOS these files will automatically be included and visible to anyone downloading the files on computers running Windows.




*[Figure 2: A screenshot of Windows' File Explorer showing a directory structure. The main directory is titled ExampleDirectory and has subdirectories titled ExampleDirectory01 to ExampleDirectory04. Within the main directory is a directory titled __MACOSX which contains duplicates of the other directories.]*



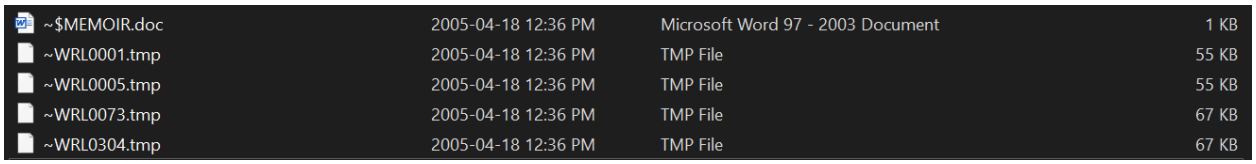

*[Figure3: A screenshot of Windows' File Explorer showing three files named ._ExamplePDF01.pdf to ._ExamplePDF03.pdf. Each of the PDF files is 1 KB in size.*

## Windows

People using Microsoft Office (or encountering files uploaded by people who do) may have found files that they didn't create and that don't seem to serve any purpose.

- ~$
    - These are files that have a filename beginning with ~$ followed by the name of an existing file and ending with common file extensions like .doc, .xlsx, or other Microsoft Office formats. These files are intended to be temporary and are often hidden by Windows' File Explorer.
    - These files are created by Microsoft Office applications as "lock" or "owner" files. They record information on who currently has a file opened so that other users can't open and edit the file at the same time. These files are intended to only exist while the specific Office application is open, although they can stick around afterward due to crashes or other errors that cause them to be forgotten by the application that created them. If you've ever been told a file you're trying to open is already opened by another user and asked if you'd like to make a copy, it might be because there's one of these files lurking somewhere.
- Thumbs.db

- These files contain data to speed up displaying files in a directory as thumbnail images. These are now generally stored in centralized directories (such as C:\Users\[username]\AppData\Local\Microsoft\Windows\Explorer) but in the past could be found in other directories.

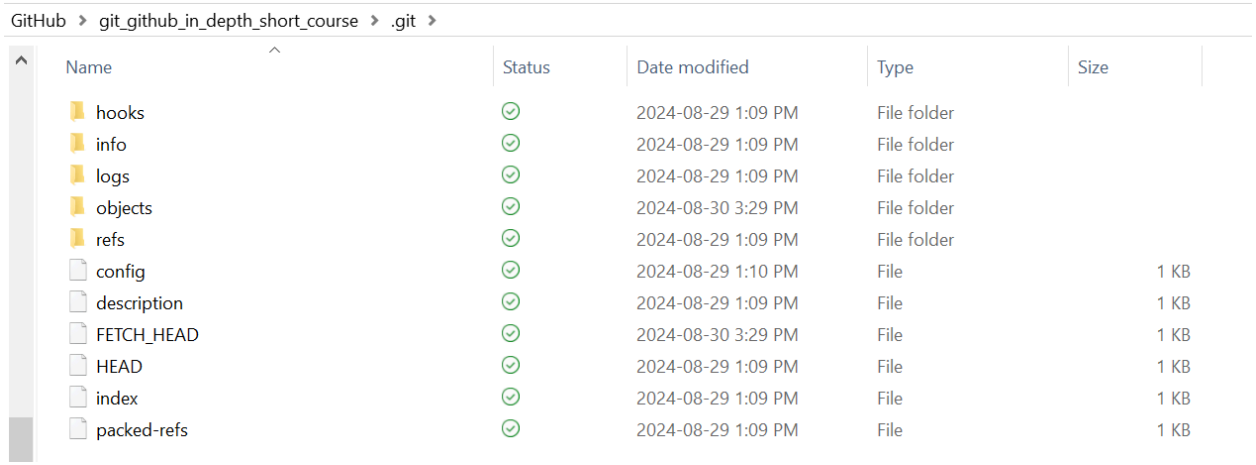



*[Figure 4: A screenshot of Windows' File Explorer showing a .doc file with a filename beginning with ~$. It's 1 KB in size. Below it are four .tmp files that have file names beginning with a ~.]*

## Other

- ~[something].tmp
  - This is an example of a temporary file automatically created by a piece of software. Many different pieces of software create temporary files or directories and may use different file naming conventions or file extensions. These may be used to improve performance, contain customization for settings, or act as backups in case of software failure. Generally, these do not need to be archived, but always check with the creators of the dataset.
- .[directory name]
  - Regardless of which operating system you're using, you may have encountered hidden directories that begin with a "." which often contain configuration files. Depending on the data you're working with, it may be beneficial to keep these directories.
- .git
  - Git uses .git directories for tracking changes and version control.[3] Generally, these will not be synced to GitHub repositories and it is advised not to share their content for privacy and security reasons[4].
- __pycache__
  - These directories are sometimes automatically generated by the Python interpreter when a Python script is first executed to speed up future executions. For archiving, these can be deleted if they're present.

[3] "What is the .git folder?" on Stack Overflow.
[4] "Is it safe to share .git folder of a public repo?" on Stack Exchange.



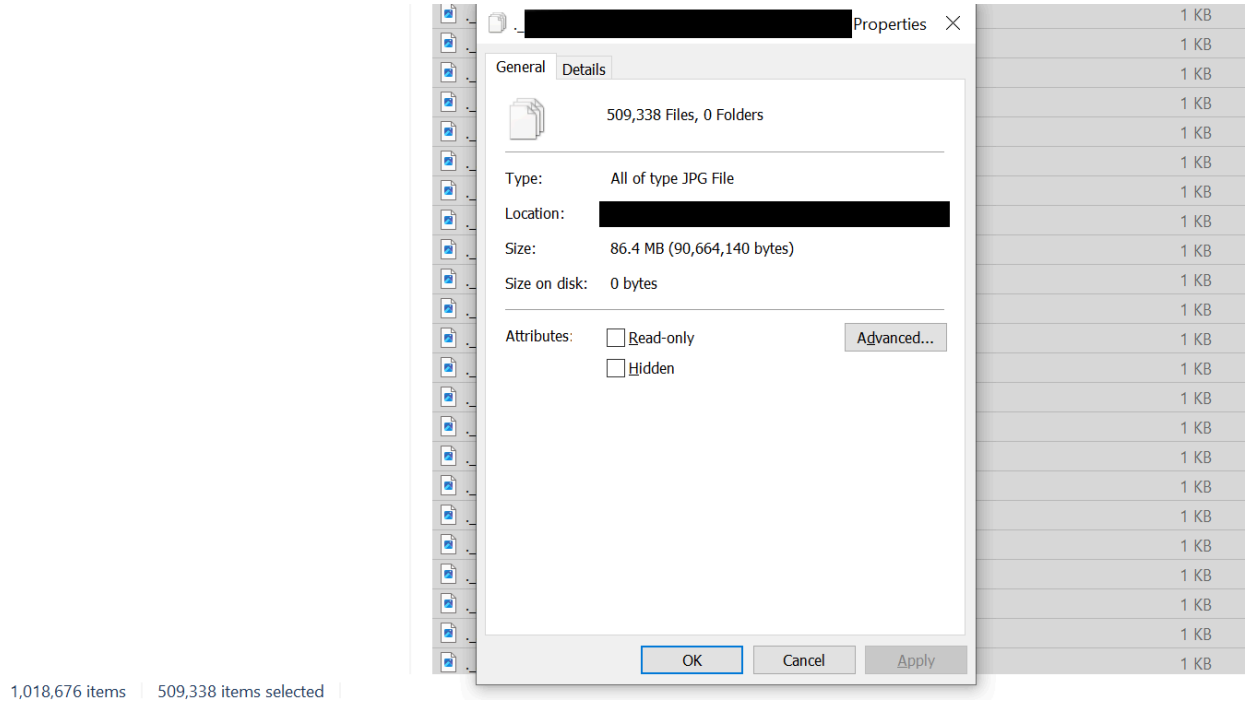

*[Figure 5: A screenshot of Windows' File Explorer showing the contents of a .git directory. There are subdirectories titled hooks, info, logs, objects, and refs. There are also files titled config, description, FETCH_HEAD, HEAD, index, and packed-refs.]*

## Why Does This Matter?

While these files are unlikely to do anything other than cause minor confusion when sharing files with colleagues, they become more problematic when they are included with datasets published in repositories.

Published datasets should include descriptions of exactly which files and directories are included in them and these CHAFF files will rarely be listed in the README file or other documentation. In many cases even if you contact the original authors of the dataset, they won't even know that these files were present! Since datasets in repositories will ideally be available in the future, when computer systems and software will have changed, there's no guarantee that people will understand whether these files are relevant or not.

Additionally, they can cause problems with accessing files within the dataset themselves. Generally, these files are small and are only a few kb in size, but in extreme cases, you can end up with thousands or even hundreds of thousands of files which can make accessing the other files within the dataset more time-consuming and difficult. When there are suddenly 500,000 extraneous additional files in a dataset doing anything with it will take more time.




*[Figure 6: Screenshot of Windows' File Explorer and Properties pop-up. File Explorer shows there are 1,018,676 items with 509,338 selected and shows a number of ._ files selected. The Properties pop-up says the 509,338 ._ files take up 86.4 MB of space.]*

# What To Do About Them

## Stopping Them

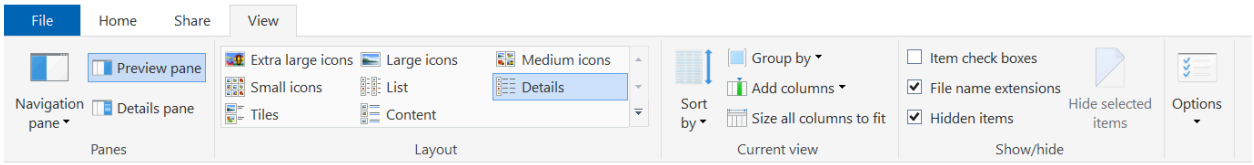
- There isn't any easy way to prevent MacOS or Windows from creating these files and you generally wouldn't want to, as they are often required for the operating systems and software to function properly. However, it is possible to create .zip files (and other archives) without including them by using the command line or software like BlueHarvest.

*Note: The below command line examples use ".zip" as the file extension, however they should work using any archive file extension.*

## Finding Hidden Files

- MacOS
    - In Finder you can view hidden files using "Command + Shift + ."; however, this ***will not*** show you .DS_Store files. Instead, you must use the command line or alternative software.
    - Command line example for the MacOS or Linux shell to examine .zip files for CHAFF inside of .zip files without extracting everything:

        - `unzip` is a command line tool for listing and extracting compressed files in archive files. It is installed by default in MacOS and most Linux distributions.
        - List files in a .zip file: `unzip -l file.zip`
- Windows
    - Open File Explorer (previously Windows Explorer) from the taskbar.
    - Select View > Show > Hidden items.




*[Figure 7: A screenshot of Windows' File Explorer showing the "View" submenu with the checkbox next to "Hidden items" selected.]*

## Deleting Them (Threshing the CHAFF)

*(The instructions below were written and tested before October 2024. Always be sure to test commands to ensure they work with your specific setup and the file formats you're working with.)*

- MacOS or Linux shell:
    - You can delete files in an existing zip archive using the zip command in the command line. The zip command supports multiple expressions with wildcards.
        - `zip -d file '*/.DS_Store' '__MACOSX/*'`
    - This command excludes .DS_Store and __MACOSX files and directories from being including in the archive when creating data.zip
        - `zip -r data.zip . -x ".DS_Store" -x "__MACOSX"`
- Windows
    - If you encounter ~$ files or other temporary files when creating an archive file, the first thing to check is if all of the relevant files are closed. Once the files and associated software are closed, many temporary files will be deleted and the archive file can be created without their presence.
    - You can open .zip files and delete any files manually.
    - The command line version of 7-zip allows you to delete specific files and directories from archives using the Windows Command Prompt without requiring you to open or extract them.
        - For example, the following commands will delete all .DS_Store, __MACOSX, and ._ files from all .zip files in the current directory:
            - `vfor /r %v in (*.zip) do C:\\[install location]\7-Zip\7za d -r "%v" .DS_Store`
            - `for /r %v in (*.zip) do C:\\[install location]\7-Zip\7za d -r "%v" __MACOSX`

- `for /r %v in (*.zip) do C:\\[install location]\7-Zip\7za d -r "%v" ._*`

### GitHub

The files mentioned in this post can also end up in GitHub repos. Thankfully, by including these file types in a .gitignore file, you can ensure that they're not uploaded.

Code Blame 9 lines (7 loc) · 111 Bytes

```
1
2     # DS store created on a mac
3     .DS_Store
4     ._.DS_Store
5     **.DS_Store
6     **/._.DS_Store
7
8     # PowerPoint temporary
9     ~$*.ppt*
```

*[Figure 8: A screenshot of a .gitignore file on GitHub. It is set up to ignore .DS_Store files and ~$&.ppt files. ]*

## Conclusion

Many other types of CHAFF Data were not included in this post and we're sure that more will be created in the future due to the exponential growth in technology. If you come across a file in a dataset that is not described in the documentation, you should always ask the dataset authors what it is and what they would like done with it. Often it will be important and will need to be included in the documentation– but other times you might be dealing with CHAFF!

*Thanks to Molly Hirst and Lubov McKone for feedback on this post!*