Version Control

Preservation and Curation of ETD Research Data and Complex Digital Objects

INSTITUTE of Museum and Library SERVICES

EDUCOPIA INSTITUTE

Photo by Rene Bohmer on Unsplash

Version Control

A year after turning in the final paper, you find a great opportunity to re-contextualize and present your work at a conference. All you need to do is use your existing research materials to address new questions, and then present the work. However, when you open the folder on your computer, all you can find is the raw data. The final version of the research materials you need is trapped in the appendix of a PDF. What started as an easy opportunity now looks like a week of sleepless nights copying, pasting, and editing files, just to get back to the starting line. ■

Rationale and Motivations – Why

Research is active and iterative. You will edit and re-edit your research materials many times before finishing your thesis or dissertation. How will you know that you are working with the most current revision of your materials? You can accomplish this through a process called *version control*.

Version control (also known as revision control or source control) is the process of managing changes to your files over time. Version control can help you to track your revision history, including when, why, and by whom changes are made.

There are many reasons to create a new version of content, including:

1. Saving a new draft of text for editing.

2. Refining raw survey results into a clean dataset for analysis.

3. Producing a transcript of an interview based on an audio recording.

4. Creating a smaller version of an image to post online.

During the research process it can be essential to return to an earlier version in order to take your analysis in a new direction, fix a mistake, or review research steps. To successfully return to an earlier version, a copy of that version needs to be stored somewhere that can be identified and accessed later.

Photo by Igor Ovsyannykov on Unsplash

Photo by Kari Shea on Unsplash

The Basics – How to Do It

Version control is all about process. This process can be manual or software-assisted, but it requires organizing the way you work.

At the beginning of a research project, it is important to create a stable folder structure in which you can organize materials. The specific folders will depend on your own research process. File organization could be based on how you plan to gather materials, which experiment or process generated them, when they were created, or other strategies. The key is to use folders that make sense to you and allow you to easily find your materials.

Manual Version Control

A simple method to designate a revision is to note it at the end of the file name. This way, files can be grouped by their name and sorted by version number. For example:

- image1_v1.jpg
- image1_v2.jpg
- image2_v1.jpg
- image2_v2.jpg
- ...

If you use version numbers, one issue that can arise is that computers will sort files based on the position of the characters. This can lead to strange, unhelpful results. For example:

- image1_v1.jpg
- image1_v10.jpg
- image1_v2.jpg
- ...

A good practice that can help you to avoid these problems is to use dates to designate version numbers. If you choose this strategy, format dates as year-month-day (20150930). Using this order will help avoid confusion when collaborating with other researchers or systems that use a day-month-year or month-day-year, and it will help your computer sort versions in chronological order. For example:

- image1_20151021
- image1_20151214
- image1_20160123
- ...

If the files you are using are created or edited collaboratively, you may want to incorporate names or initials into your file naming conventions so that you know which versions contain updates by each individual on your team. For example:

- dataset1_20160402_KES
- dataset1_20160301_WTC
- dataset1_20160814_GSC
- ...

Software-Assisted Version Control

More powerful methods of version control include the use of software tools like Git and Subversion. These tools store your content in such a way that they can remember its state from revision to revision. Usually, these tools allow you to check your content in and out, ensuring that revisions never happen simultaneously in two different locations (e.g., if two researchers working together on a project both attempt to revise the same file at the same time). Key differences between these software-assisted methods and the manual methods described above include the following:

1. You can only view and edit the working version of a file.

2. When you change a file, you can save a revision and attach a short summary of your changes.

While version control software was developed by coders for coders, it can be used to manage any kind of data. However, using these tools does require an investment of time to learn how to use them properly. Extensive guides, tutorials, and classes for these version control tools are available for free online (see the **“Resources”** section for some initial leads).

Photo by Matus Kovacovsky on Unsplash

Tools – What to Use

Manual Version Control

Managing versions manually only requires using a file manager, whether that is the File Explorer in Windows, Finder in OS X, or Nautilus in Ubuntu Linux. These file managers also exist in Dropbox folders, the Google Drive interface, and many other storage systems that may be local to a machine or cloud-based.

Learning to use keyboard shortcuts can make the process much faster and easier. For example, to create a new version of an existing file, select the file in the file manager, then copy and paste it using **ctrl+c** and then **ctrl+v** in Windows and Linux, or **cmd+c** and then **cmd+v** in OS X. To update a version number in a file name, use **F2** in Windows and Linux, or **"enter"** in OS X.

Software-Assisted Version Control

If you choose to use software-assisted version control, there are a few available options, including Git and Subversion. A very useful feature of systems like Git and Subversion is the ability to automatically host a repository of versions with an online service such as GitHub or BitBucket. This makes it easier to share materials and their versions with collaborators. Your choice of service will depend on the specific requirements of your research workflow.

Resources

All links provided were last checked 10/3/2017, and the content we reference here was saved on that date in the Internet Archive's Wayback Machine. For links that no longer work, please visit the Wayback Machine and enter the url to surface the resource.

- The digital humanities center MATRIX at Michigan State University has provided advice on how to structure file names based on its experience with oral history projects, but this guidance is broadly applicable to general research processes.
- Udacity offers a free online course on how to use Git and GitHub with interactive exercises to familiarize you with using the tools.
- Another helpful GitHub guide is available from Hello World.
- The Subversion community provides free access to the book Version Control with Subversion in both html and pdf formats. The book was authored by some of the Subversion software developers.

Photo by Melanie Croce on Unsplash

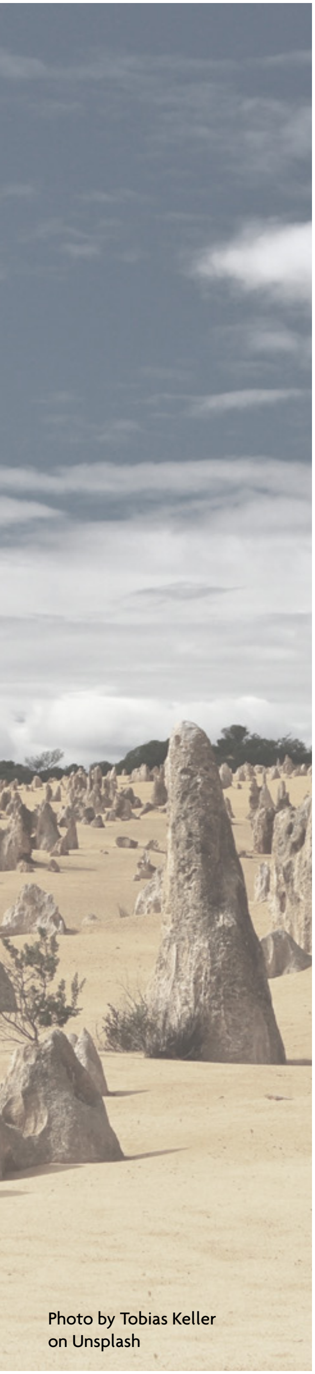
Photo by Tobias Keller on Unsplash

Activities

1. Find a folder of research materials that you have collected on your computer. Look through the materials and answer the following questions:
   a. Are there multiple versions of the same materials (documents, images, etc.)?
   b. How are the different versions labeled?
   c. Can you quickly identify a file's most recent version? Its authoritative version? Its original version?
2. Create a diagram of your research process. Do this however you prefer, with software, with pen and paper, etc. Locate the materials that you are creating, collecting, and editing as part of your research process and answer the following questions:
   a. Will your research materials be edited or transformed between the time that you collect them and the time you publish them in your thesis or dissertation? If so, is it important to save the intermediate versions? What conventions might you use to track the versions?
   b. Do you expect to combine multiple files into a single file (e.g., editing multiple clips into a single video)? If so, how do you plan to manage the relationships between the original objects and the combined object (e.g., naming conventions, folder context, readme files, etc.)?

INSTITUTE of Museum and Library SERVICES

EDUCOPIA INSTITUTE