Chapter 2

WF Chapter 2: Extended directory naming conventions

This is a rough working draft.

As of 2007-07-11, I use a somewhat more elaborate set of special purpose directories than described in *Chapter 2*. In part, this more elaborate scheme reflects how much data analysis I do. But, it is also more elaborate than I actually need. With time, I suspect it will evolve to the structure described in the book.

Other special purpose directories

Over the years, I have found it useful to have utility directories that are common to almost every project folder. I have gotten into the habit of starting these directories using special characters (e.g., `!`, `@`) since this will move them to the top of the list of folders when I look at them in my file manager. If you want to avoid these characters, you could replace them with an underscore `_`. Here are the directories along with brief descriptions.

\\`$_History` is a directory which includes information on the history of the project. This is quite handy when you come back to a project years later.

\\`!_To_do` contains work to be completed that has not been started. (Work in process goes in the \\`Work` directory). Files in this folder are essentially a to-do list. If I think of something that needs to be done, a reprint I need to read, a prior do file that needs to be revised, etc., I put it here until I get a chance to do it.

\\`#_Archive` holds completed analyses and drafts that have *already* been archived at another location. By putting them in this directory, I remind myself that these files no longer require archiving. Details on archiving and backing up are given in Ch. 8-Save. As subdirectories, I use directories with dates in a year-month-date format that sorts by date:

\\`2006-01-12` are files archived on this January 12, 2006.

\2006-02-09 and so on...

\%_To_clean is where I put those inevitable files that I have lost track of. These files need to be examined and and moved to their proper location. When I am in the process of archiving data, I might move entire directories here to remind myself that I need to finish documenting and archiving them.

\@_Backups is a folder with *short-term* copies of files that I have copied just in case I accidentally delete something or change that causes a problem. For example, suppose that I have written a series of do files to create scales, select cases, merge data sets, and so on. These programs seem to work, but I have not been as careful as I want to be in adding labels and comments. I might copy the current versions here before I start revising the programs. If my improvements aren't working, I can go back to the prior version of the files. Ch. 8-Save discusses the distinction between backups which are short-term and temporary and archives which are long-term. As with archives, the \@_Backups folder might have subfolders with the date on which the backup was made, such as:

\2006-02-12 for backups made on February 2, 2006.

Or, I might use names corresponding to what the backups are for. FOr example,

\VarConstruct for files related to variable construction.

~_Hold_then_delete is like the Recycle Bin in Windows. Files go here when you are ready to delete them. If you "deleted" them by mistake, you can easily recover them. But, when the project is over, you can feel confident that if you delete everything in this folder, you haven't deleted anything critical. For organizing your files, it is important both to make sure that files you need are placed where they can be found and to keep track of files that you don't need. If you don't keep track of files that can be deleted at the end of the project, you will end up with lots of files that you don't know what to do with (sound familiar?). When you need disk space or the project is finished, this directory is deleted.

Now that I've explained the purpose of various special folders, I can tell you another reason why I use special characters to indicate special purposes. Suppose that I have finished a draft of a paper and it is being sent out for review. This is a good time to go through the files I have and make sure everything is in order. It will be much faster to do it now than when I get the reviews back and I have forgotten many of the details. I might start by renaming each folder I need to clean with a prefix %. For example, I would rename \Text to \%Text and change the name back after I have cleaned out the files in that directory.

2.0.1 Remembering what directories are for

You need a way to keep track of what a directory is for and what goes where. I have found several tricks that help. First, decide on a structure you like and use it for everything you do. Eventually, it will become second nature. For example, if every project directory contains a subdirectory \Work, you will know where to begin your work when you return to the project. (Of course, you can chose a different name than \Work for your "work in progress". The key is to use the same name for all of your projects.) Second, I use particular starting symbols for a directory name to remind me of what the directory is for. For example, any directory that starts with a % needs to be cleaned up and then archived. For example, if I have a subdirectory \Add_panel4 with programs I just completed that add the recently released panel 4 to my master dataset, I would rename it `\%Add_panel4` when the project is done and I need to verify the files, back them up and move them to an archive. This serves as a quick reminder of what needs to be done to these. Here are the conventions I use:

Starting Symbol	*Indicates*	*Example*
`-`	Empty directories used as labels.	`\- Submitted draft of paper`
`%`	Files that need to be checked and cleaned.	`\%Ch3Examples`
`@`	Backups that haven't been archived yet.	`\@2006-06-20`
`~`	Scratch files that can be deleted if space is needed.	`\~testdofiles`
`#`	Archives already located on a secure remote site.	`\#2006-06-20-draft1`
`_`	Private directories on collaborative projects.	`\_Scott_Long`

Third, pick names for directories that are short (unless you don't mind typing long names) and informative. You can also add `read.me` files in each directory that contain information about what is in a given directory. For example, the `read.me` file might contain:

```
Project:    Workflow of Data Analysis
Directory:  \WorkFlow\Text
Content:    Text files for workflow book.
Author:     Scott Long
Created     2Feb2006
```

The problem I have with `read.me` files is that you have to open them, so they aren't as useful for a quick reminder. They can also end up in the wrong location and thus are more confusing than helpful. If you want to use a file to document what is in a directory, consider giving it the directory name with the suffix `is`. For example, the "read me" file for the `\Workflow\Text` directory, would be called `Workflow_Text.is`.

Rather than a `read.me` or *directory*`.is` file, I prefer to use *naming directories* whose sole purpose is to remind me what is in the directory above it. Here is an example for the directory `\_Scott_only`:

```
\_Scott_only -

    \- Scott Long private directory
```

Or,

```
\#_Archive

    \#2006-01-12

        \- Archived to SGEX_2006_02 at IU
```

The *naming directory* `\- Archived to SGEX_2006_02 at IU` tells me what `\#2006-01-12` is for, but also reminds me where the archive itself is physically located.