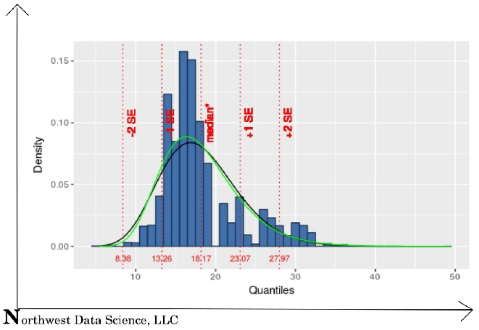


Northwest Data Science, LLC

To: Jeff Hardy

From: Bruce Marron

Subject: Data Management Architecture, Part 2

Date: July 31, 2018

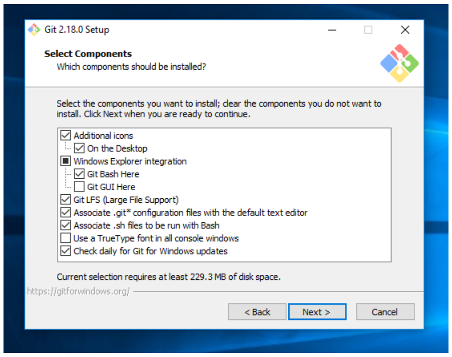
Hi Jeff! This memo documents the proposed front-end of the data management architecture: the ability to create a collection of datasets for subsequent analysis from the total universe of proprietary datasets. The process is depicted in Figure 1 below.

Rather than try to set up a complicated "Library of Congress"-style tracking system for the proprietary datasets, we can use the power that is provided by Linux-based utilities to search both for and within ASCII .csv files. This means you can name an original dataset whatever you like as long as the moniker is unique. We might consider adding a simple date to the .csv file name (in yyyy/mm/dd format). As we know, the total universe of proprietary datasets is expected to grow over time so we want a very flexible system. The proposed system is an analog of the hard copy storage system at the new, ultra-modern James B. Hunt, Jr library at North Carolina State University. Printed materials are literally thrown into giant bins without regard to content. Everything is completely mixed up! It's the sorting algorithm that is used to find and pull a book. Check it out (the actual printed storage facility is at 40 to 51 seconds in the video): library

To make this a possibility, we'll install the "Git for Windows" package in our Windows boxes and supplement the installation with a whole suite of Linux-based utilities obtained from the CygWin collection (https://cygwin.com/). Installing "Git for Windows" provides the Git version control system (software) for tracking changes in computer files. The suite of Linux-based utilities from CygWin will make searching, sorting, and collecting .csv files possible with the utmost precision.

A. Install Git 2.18.0

1. Go to the "Git for Windows" website (left-click here)
   a. Click on the "Download" button to download, `Git-2.18.0-64-bit.exe`
   b. Run the `Git-2.18.0-64-bit.exe` installer accepting defaults EXCEPT
      i. Check daily for Git Windows updates



Select Components

Which components should be installed?

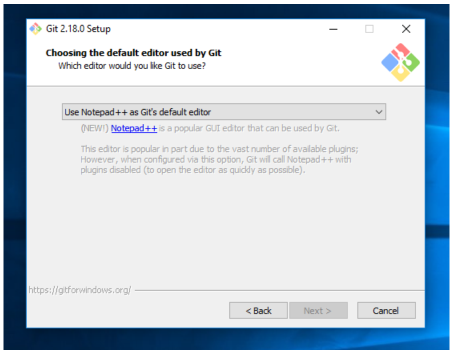
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Additional icons
  - ☑ On the Desktop
- ■ Windows Explorer integration
  - ☑ Git Bash Here
  - □ Git GUI Here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Use a TrueType font in all console windows
- ☑ Check daily for Git for Windows updates

Current selection requires at least 229.3 MB of disk space.

https://gitforwindows.org/

ii. Use Notepadd++ as Git's default editor (We'll install Notepad++ in a moment.)



Choosing the default editor used by Git

Which editor would you like Git to use?

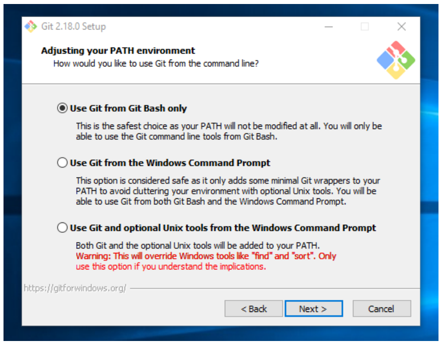
Use Notepad++ as Git's default editor

(NEW!) Notepad++ is a popular GUI editor that can be used by Git.

This editor is popular in part due to the vast number of available plugins; However, when configured via this option, Git will call Notepad++ with plugins disabled (to open the editor as quickly as possible).

https://gitforwindows.org/

iii. Use Git from Git Bash only



Adjusting your PATH environment

How would you like to use Git from the command line?

● **Use Git from Git Bash only**

This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

○ **Use Git from the Windows Command Prompt**

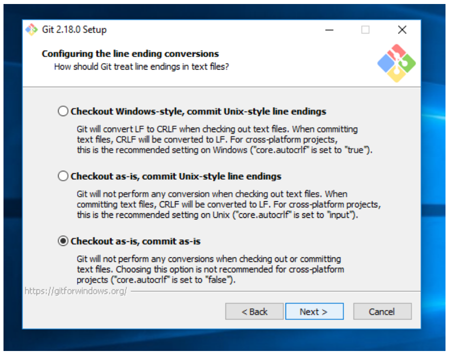
This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.

○ **Use Git and optional Unix tools from the Windows Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/

iv. Checkout as-is, commit as-is



Configuring the line ending conversions

How should Git treat line endings in text files?

○ **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

○ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

● **Checkout as-is, commit as-is**

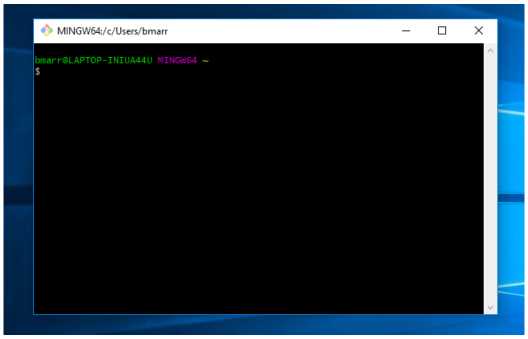
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

https://gitforwindows.org/

c. Upon completion of the installation a GitBash icon should appear on your Desktop. If convenient you can leave it there or pin it to your Taskbar.

d. Double-click on the icon to open Git Bash

   i. A new, command line window will appear



```
bmarr@LAPTOP-INIUA44U MINGW64 ~
$
```

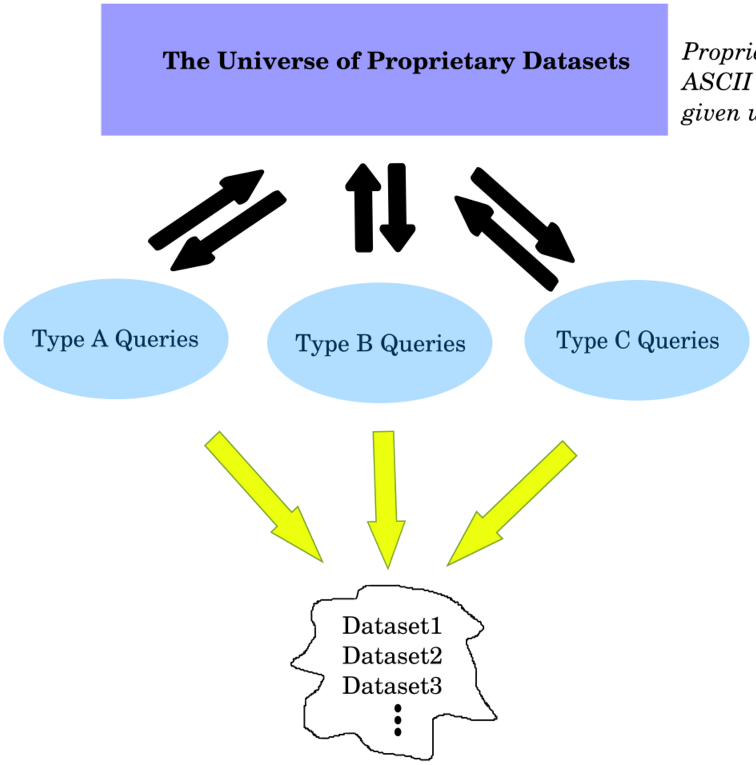
e. We will use this interface to run the Git version control system and to rapidly find proprietary datasets.

B. Supplement Git Bash with Linux utilities

1. Unpack the `LinuxUtilities_Windows.zip` file
2. Use File Explorer in Windows to locate the `\bin` directory of the Git installation (`C:\Program Files\Git\bin`)
3. Copy and paste all of the individual files from `LinuxUtilities_Windows.zip` into `C:\Program Files\Git\bin`. Note that these are mostly .exe files.

C. Install Notepad++ 7.5.8

1. Go to the Notepad++ website (left-click here)
2. Scroll down until you find the link, "Notepad++ Installer 64-bit x64"
3. Click on the link to download the Notepad++ installer, `npp.7.5.8.Installer.x64.exe`
4. Install accepting all defaults



Proprietary datasets are stored as ASCII text files in .csv format and are given unique monikers.

Type A queries search for collections of datasets by queries of the file name.

Type B queries search for collections of datasets by queries of key words in the fields of the file header.

Type C queries search for collections of datasets by queries of the data fields.

All queries are done from a BASH command line (in either Linux or Windows) using regular expressions with a variety of Linux-based utilities including "sed" and "grep."

The collection of datasets is then imported into R for subsequent Tyoe I and Type II queries.

Figure 1: Proposed data management architecture, Part 2. Building collections of proprietary datasets for subsequent analysis.