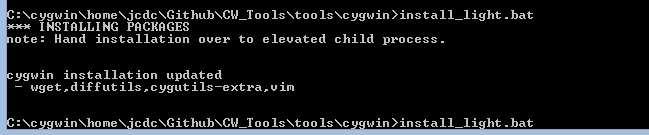
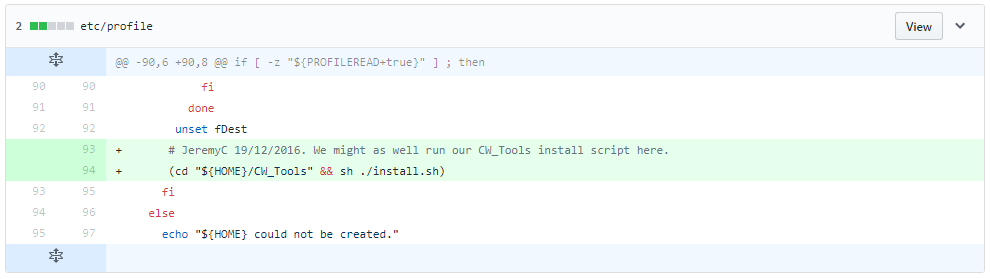
**Create Cygwin\_Light for CW Tools**

**1) Run install\_light.bat to download and create (in one go) an installation of Cygwin in the local directory named after today’s date, i.e. “Cygwin\_Light\_DDMMYYYY”.**

  
  
  
  
**2) Edit etc/profille to call the install.sh that comes with CW\_Tools.zip:**  
  
For easy cut-and-paste:  
(cd "${HOME}/CW\_Tools" && sh ./install.sh)  
  
  
  
**3) Add the Cygwin installation to Github, as a new repository named “Cygwin\_Light\_22042018”.**  
Create the new repository named Cygwin\_Light\_22042018 using the Github UI.  
  
Then add our Cygwin installation to the Github repository:  
  
cd Cygwin\_Light\_22042018  
git init  
  
From <https://help.github.com/articles/dealing-with-line-endings/>:

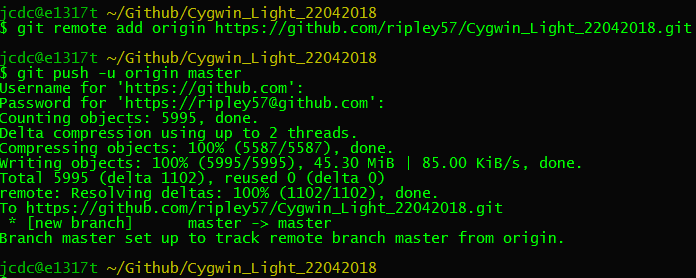
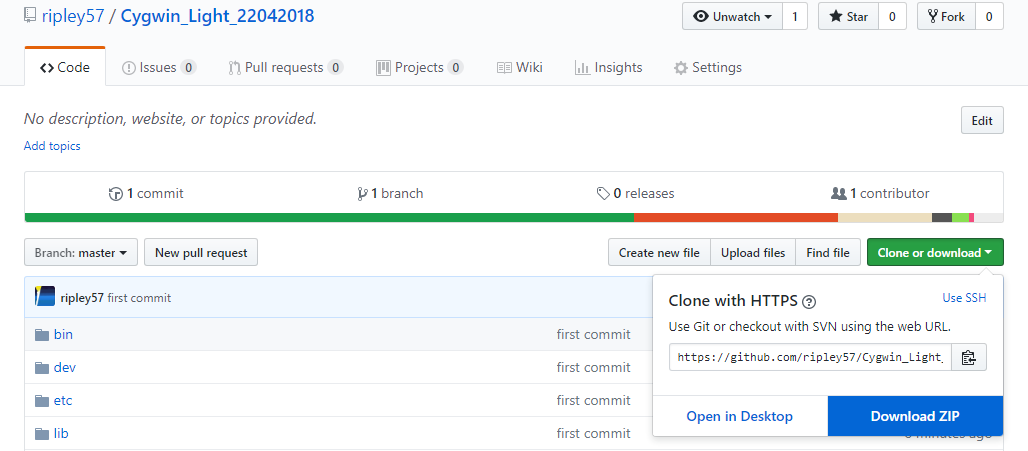
|  |
| --- |
| *“Optionally, you can configure the way Git manages line endings on a per-repository basis by configuring a special .gitattributes file. This file is committed into the repository and overrides an individual's core.autocrlf setting, ensuring consistent behavior for all users, regardless of their Git settings. The*.gitattributes*file must be created in the root of the repository and committed like any other file. Example entry:*  # Set the default behavior, in case people don't have core.autocrlf set.  \* text=auto *“Git will handle the files in whatever way it thinks is best. This is a good default option.”*  *Different options can be specified for different file extensions – see* [*https://help.github.com/articles/dealing-with-line-endings/*](https://help.github.com/articles/dealing-with-line-endings/) *ME: An alternative, that might be best for me, for all projects, is this:* \* binary *”Git will understand that the files specified are not text, and it should not try to change them.”* |

git add .

git commit -m "first commit"

git remote add origin https://github.com/ripley57/Cygwin\_Light\_22042018.git

git push -u origin master

**Note:** The last step will prompt for Github credentials:  
  
  
  
  
**4) Download the new cygwin.zip from Github and copy over the existing zip file in Dropbox.**This will download a file named Cygwin\_Light\_22042018-master.zip.

Rename this to Cygwin.zip and copy over the existing file in Dropbox.  
**Note**: We copy over the existing file in order to prevent us having to create a new Dropbox public link, which would then require us to update the link in cwtools.properties and rebuild cwtools.bat again.  
  
  
  
**5) Update CWTools.java in the CW\_Tools Github repository and rebuild cwtools.bat (and cwtools.jar), to include the new Github repository extracted directory name.**  
   
The extracted zip file will create the following directory:  
Cygwin\_Light\_22042018-master  
  
We need to add this name to the list of recognised directory names in CWTools.java:  
  
private enum Cygwin {

GITHUB ("url.github.cygwin","Cygwin\_Light-master.zip","Cygwin\_Light-master","Cygwin.zip"),

DROPBOX ("url.dropbox.cygwin","Cygwin.zip","Cygwin\_Light\_22042018-master:Cygwin\_Light-master","Cygwin.zip");  
  
Now we need to rebuilt cwtools.bat.  **Note:** Be sure to use build\_jar.sh, as this compiles the Java source files using Java 6. This means we can then run the cwtools.bat on systems with Java 6 (and newer) installed. Copy the new cwtools.bat to Dropbox.  
  
  
  
**Useful references:**  
- Get Cygwin setup-x86.exe:

<https://cygwin.com/setup-x86.exe>

- Cygwin Setup-x86.exe command-line options:  
<https://www.cygwin.com/faq.html#faq.setup.cli>

- How to determine which Cygwin package a file (e.g. banner.exe) is in:  
<https://cygwin.com/cgi-bin2/package-grep.cgi?grep=banner.exe&arch=x86_64>  
  
- Full list of Cygwin packages:  
<https://cygwin.com/packages/package_list.html>

- List installed packages and versions:  
cygcheck -c

*JeremyC 22-04-2018*