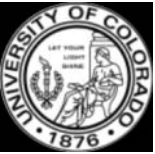


# Version Control in Embedded Systems with Subversion

Embedded System Design – ECEN5613



# Why Use Version Control?

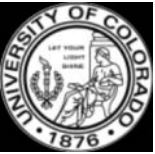
---



## SpaceX Demo Flight 2

- Incorrect propellant utilization file loaded into engine computer
- Resulted in:
  - Lower Thrust
  - Lower Trajectory
  - Lower Velocity second stage
  - **Failure to reach orbit**

Fig. 1 - Credit: SpaceX ([http://www.spacex.com/photo\\_gallery.php](http://www.spacex.com/photo_gallery.php))



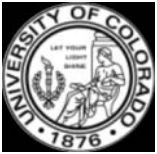
# Popular Free Version Control Systems

---

- CVS – Concurrent Versioning System
  - Released June 23, 1986
  - Original Purpose
    - Three programmers
    - No common schedule
    - Needed to work together
  - Grew into a widely-used version control program
- SVN – Subversion
  - Development started in 2000
  - Created to be replacement of CVS<sup>1</sup>
  - Fixed bugs, and added features lacking in CVS<sup>2</sup>
  - Used by SourceForge.net, GCC, KDE, Google Code, and more<sup>3</sup>



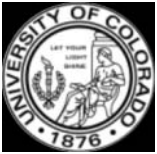
Fig. 2 - Credit: Subversion Software (<http://subversion.tigris.org/>)



# SVN Version Control Methodology

---

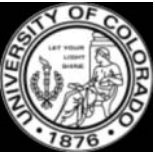
- Server – Client Based System
  - Most up to date version kept in server-side repository
  - Repository only records changes to files
  - Repository is also compressed
    - Yields very small Subversion repositories<sup>1</sup>
  - Client works off of downloaded copy of repository
- How it is used
  - Projects are “Checked Out” from server repository, creating local copies<sup>2</sup>
  - Already local copies are “updated” before any modifications
  - Project is committed after modifications are made
  - Each commit increments project version number, file version number



## How SVN Works

---

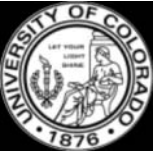
- SVN allows any number of checked-out repositories
  - Each can be modified concurrently
  - 'Locking' user out of specific files is not required
- So how are file conflicts avoided?
  - The *copy-modify-merge* solution
    - Allows completely parallel development
    - Prevents having to wait for a file
    - But Conflicts occur when:
      - Text files have overlapping modifications
      - Binary file modified by multiple users
        - No way for computer to know correct resolution
        - User must manually merge file



## How SVN Works – Copy-Modify-Merge

---

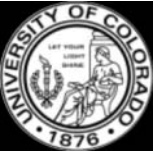
SVN	Algorithm
Users 1 and 2 Checkout / Update.	Copy
Users edit source files, schematics, etc	Modify
User 1 commits to repository	Merge
User 3 updates from repository	Copy
User 2 commits to repository, only conflicts possible with user 1.	Merge
User 3 commits to repository, only conflicts possible with user 2.	Merge



## Final Words

---

- Version Control speeds up development, eliminating the need for serialized project development
- Can save hours of development and debugging hassle
- SVN is particularly hassle free:
  - Copy-Modify-Merge nearly never has conflicts, even on large projects
  - Allows developers to always have the latest from everyone on a development team



## SVN and You!

---

- It is easy to set up *your very own* subversion repository using the ECE Student Server.
- Below are the step-by-step instructions for setting up your repository on eces-shell.





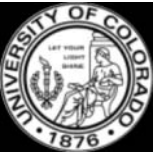
## Creating the SVN on ECES-SHELL

---

- Connect to `eces-shell.colorado.edu` using your preferred SSH client

```
svnadmin create --fs-type fsfs esdsvn
```

- Good work, that is it!
- You are now ready to add files to your svn
- You can do this from a Linux command line or with Windows graphical programs.

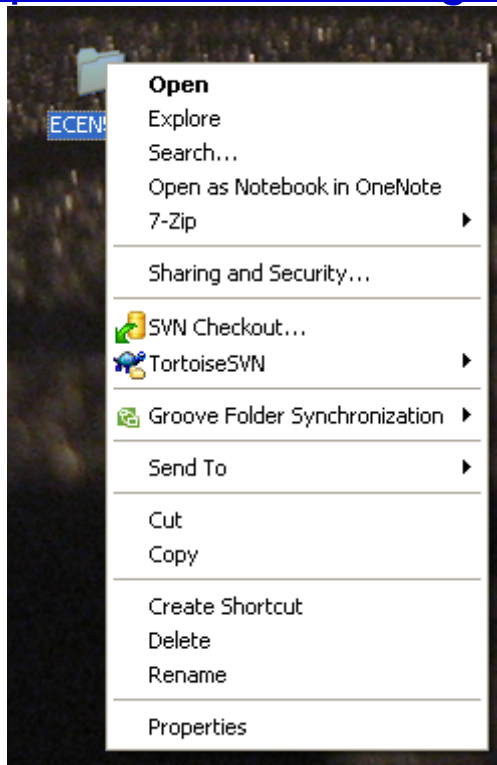


## Accessing your SVN in Windows GUI

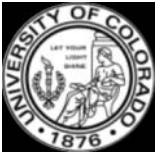
---

- Download and install TortoiseSVN from:

<http://tortoisesvn.tigris.org/>

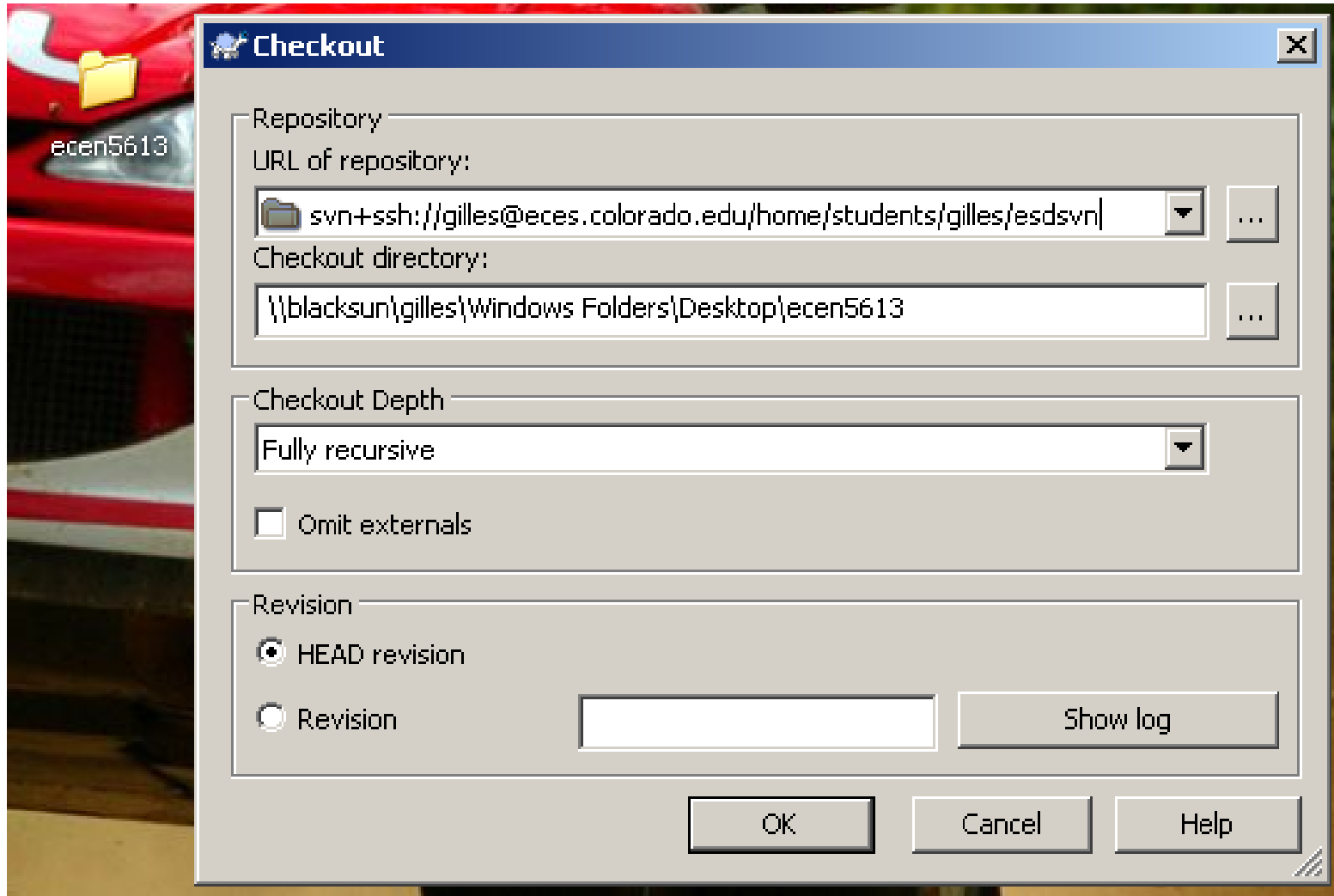


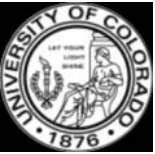
- Notice TortoiseSVN integrates into Windows



# Accessing your SVN in Windows GUI

- Check out your repository





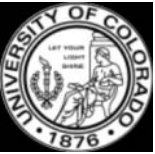
## Accessing your SVN in Windows GUI

---

- It will ask if you trust the server.
- Say Yes (or press enter)



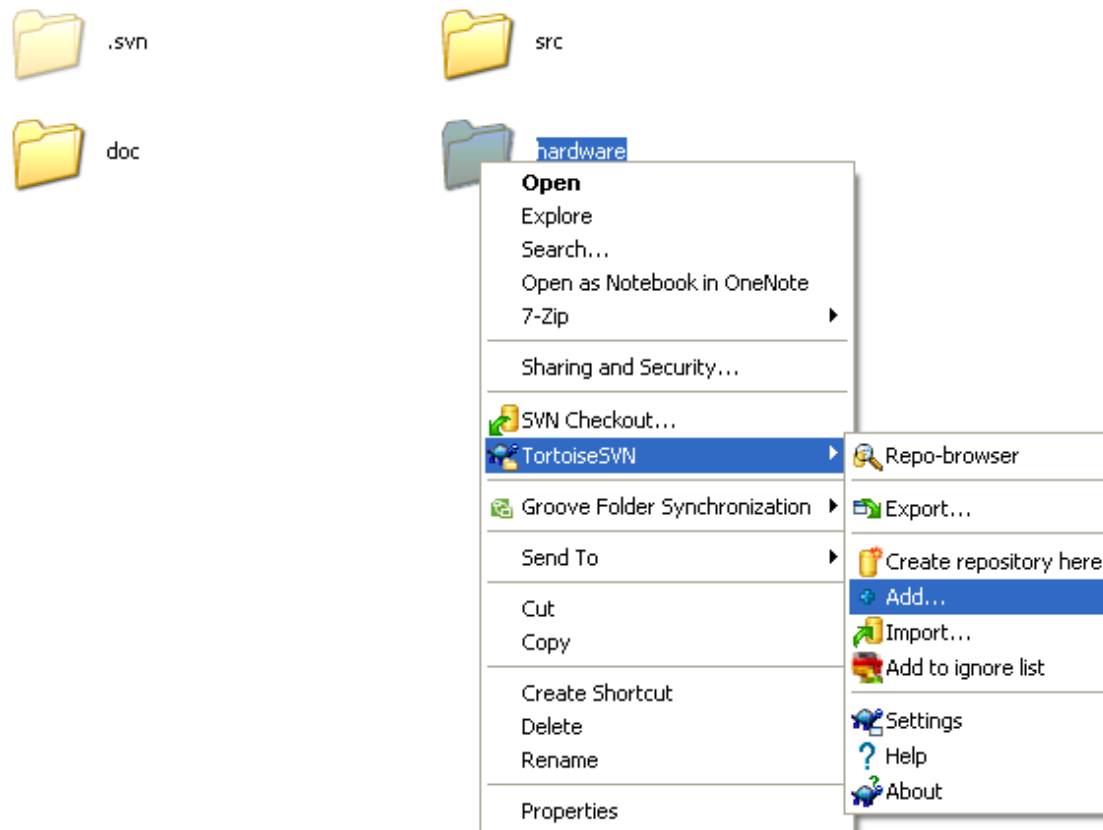
- No password is required in ECE labs.
- If not in an ECE lab, you will have to enter your password three times. This happens during checkout only.

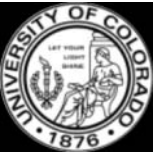


## Accessing your SVN in Windows GUI

---

- Add files to your SVN. Be cautious. Only add files you want version controlled.

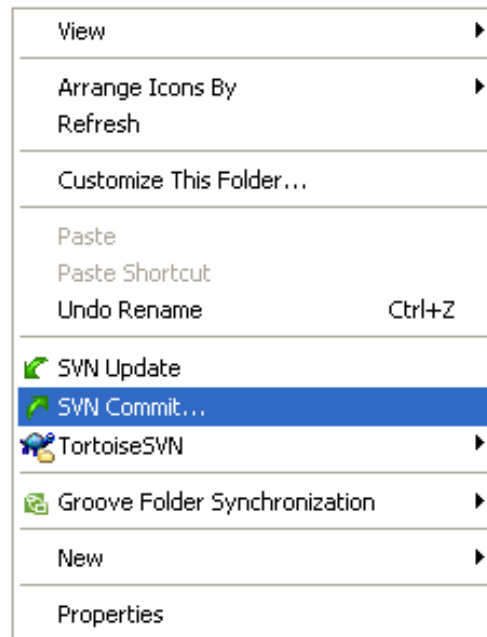
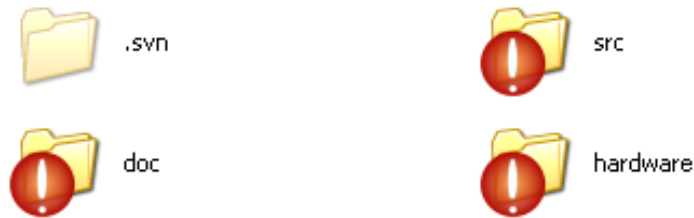


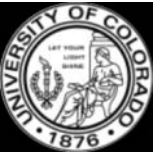


# Accessing your SVN in Windows GUI

---

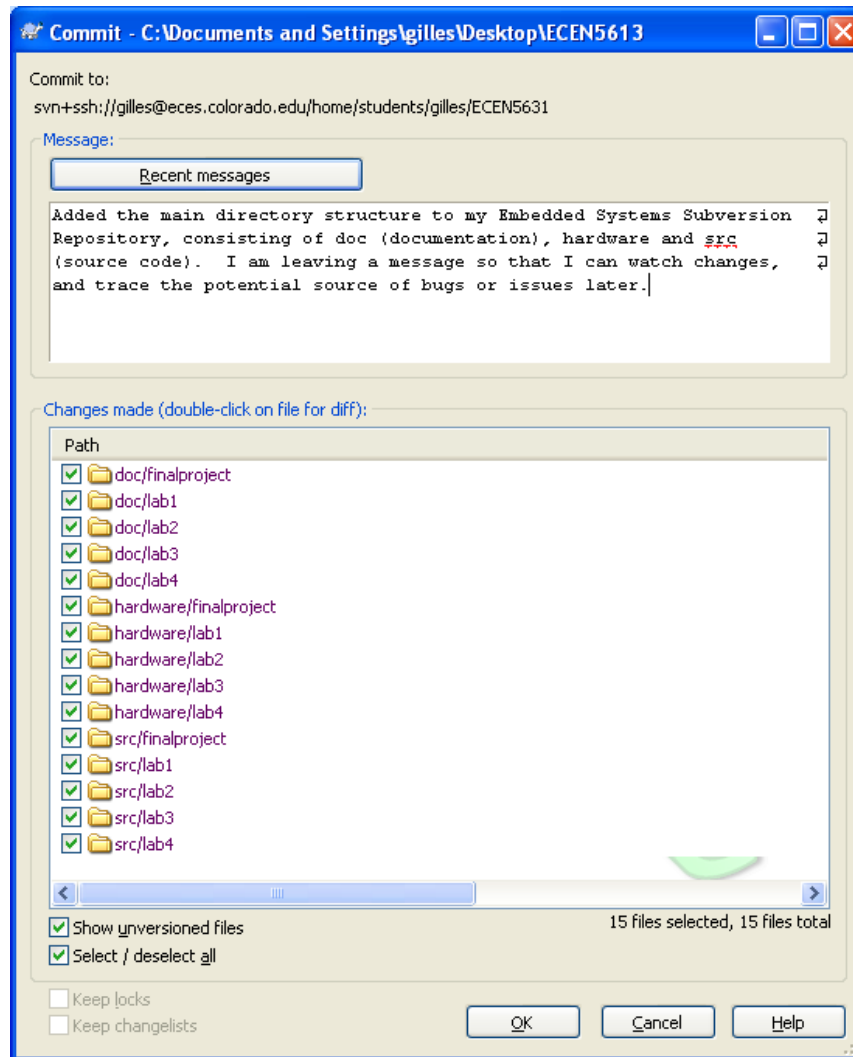
- Commit your changes to the server.

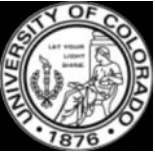




# Accessing your SVN in Windows GUI

- Leave a message so you know what you changed.

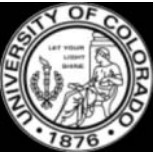




## Using Your SVN

---

- Update before you start working
  - Commit periodically as you work or when you finish
  - Leave messages when you are committing
  - You will be happy with yourself later if you leave messages
- 
- If using Linux to access the svn, run 'svn help' to learn how to do the equivalent of the above.



## References

---

- The Subversion (SVN) material in this presentation is based on information learned from three sources:
  - 1 – the free Subversion book: <http://svnbook.red-bean.com/>
  - 2 – the Wikipedia article on Subversion: [http://en.wikipedia.org/wiki/Subversion\\_%28software%29](http://en.wikipedia.org/wiki/Subversion_%28software%29)
  - 3 – Use of the TortoiseSVN SVN client: <http://tortoisesvn.tigris.org/>
- The Concurrent Versions System (CVS) material in this presentation is based on information learned from the Wikipedia article on CVS:
  - [http://en.wikipedia.org/wiki/Concurrent\\_Versions\\_System](http://en.wikipedia.org/wiki/Concurrent_Versions_System)
- The SpaceX material in this presentation is from the SpaceX “Demo Flight 2 Review Update” for the Falcon 1 Launch Vehicle, available at:
  - <http://www.spacex.com/F1-DemoFlight2-Flight-Review.pdf>
- Figure 1: Credit: SpaceX: [http://www.spacex.com/photo\\_gallery.php](http://www.spacex.com/photo_gallery.php)