

Version (Source Code) Control

SWEN-250

Overview

Motivation – why is version control useful?

Key concepts

Variations on the basic theme

Example version control systems

Motivation

Progress, far from consisting in change, depends on retentiveness. Those cannot remember the past are condemned to repeat it.

- George Santayana

Motivation Scenario: I *think* this will work

Often we want to try out a change

- Trying a new algorithm or data structure
- Reorganizing code for clarity
- Experimenting with a half-cocked idea
- Seeing if the language works like you expect

It's a lot easier if you can perform such experiments *confidently*.

- That is, you can get back to where you started
- VCS can provide a virtual trail of breadcrumbs
- If you botch things, you can return to a stable state

Motivation Scenario: How did I get here?

Like waypoints on a GPS system.

- Allows you to track progress
- You can see how your program evolved and grew, step-by-step
- You can see where you made mistakes and how long it took you to find and fix them

The fancy software engineering term is *traceability*.

- Important for scheduling, tracking, and planning
- Allows you to go back to a previous version (“hey, what did we ship for version 1.5.2.9.5?”)

Motivation Scenario: Reconstruct the Past

Teacher: “So, show me – what was the code like before you made this change?”

You: “Duhhhh.”

Motivation: Pragmatic Programmers

Tip 23

Always Use Source Code Control

The Gods have spoken.

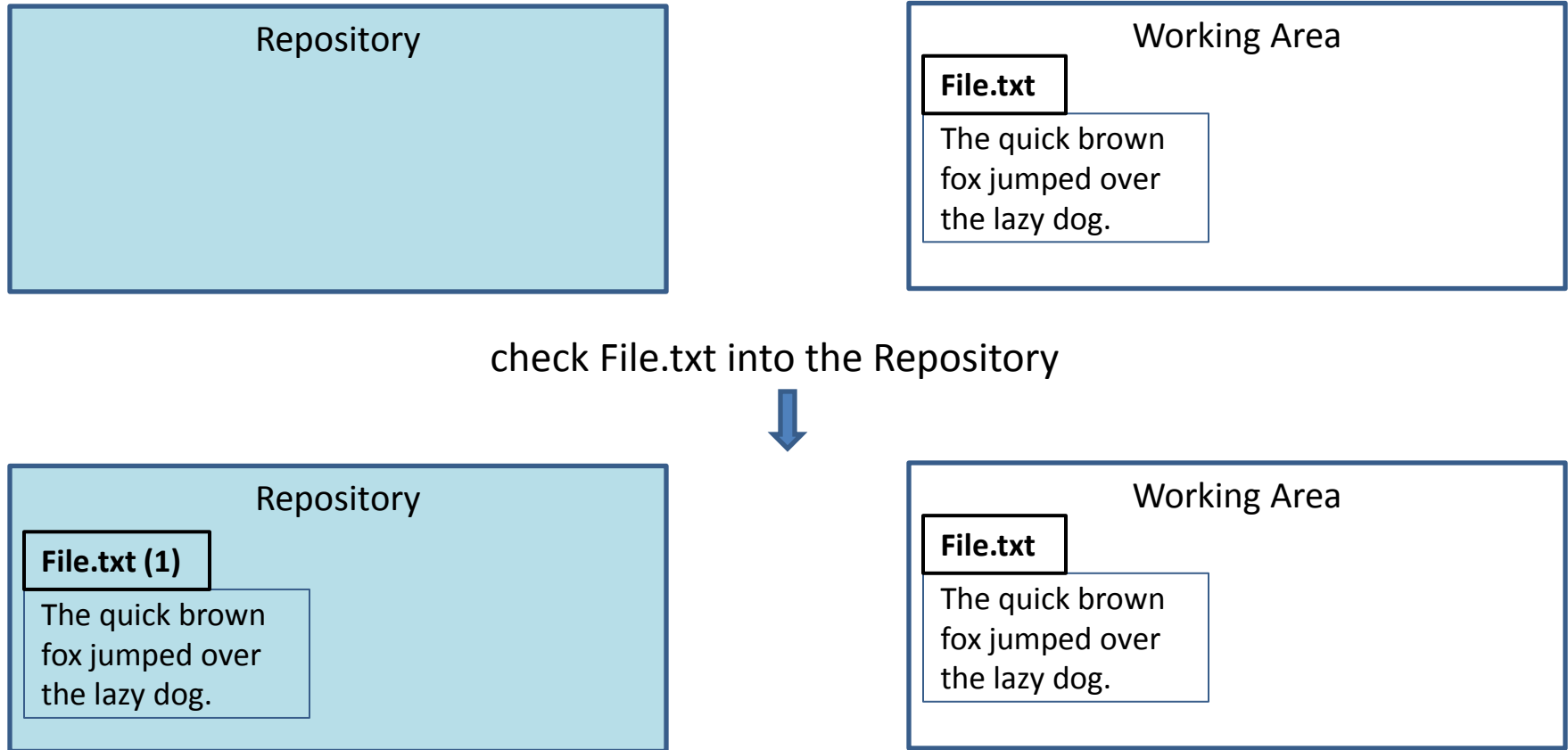
Key Concepts

A **repository** is a designated disk location (directory) where the files and “breadcrumbs” for a project are kept.

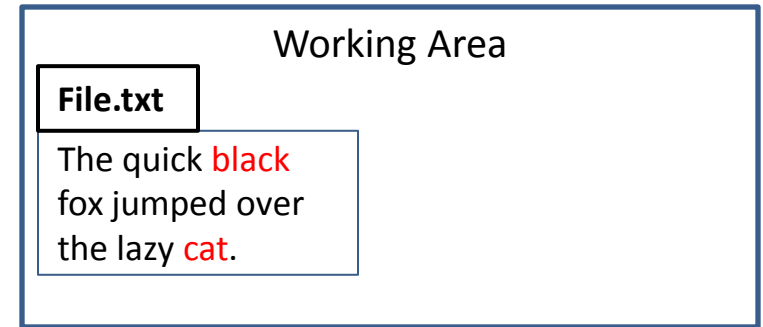
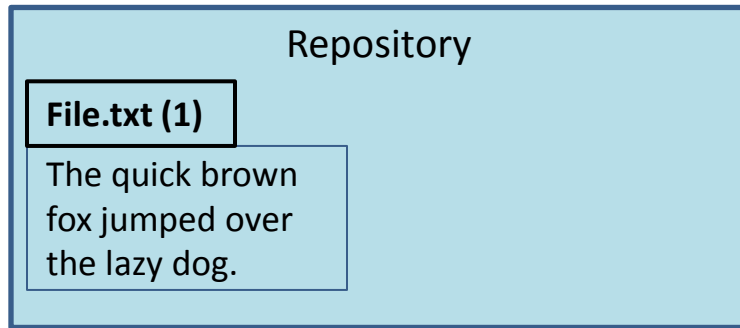
The **work area** is a disk location (directory) where the activities of editing, compiling, testing, etc. take place. Files are periodically **checked in** to the repository from the work area, creating a new version.

Files can be **checked out** of the repository (to start work on existing project, or to restore the state of the work area to a previous state).

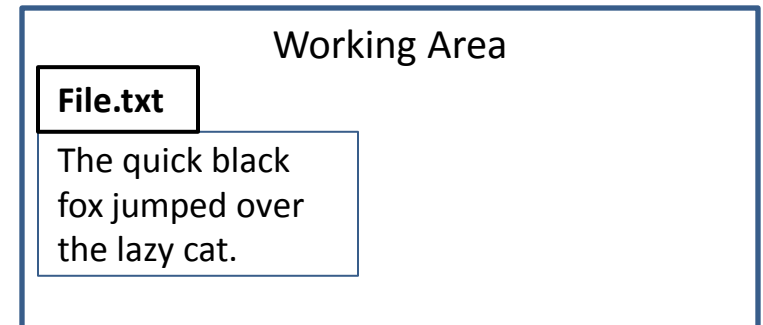
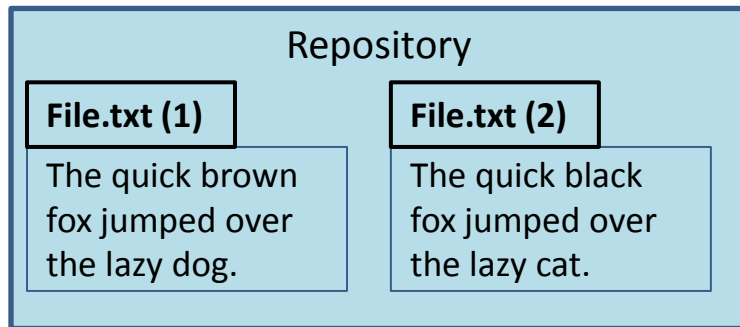
Check a File into a Repository



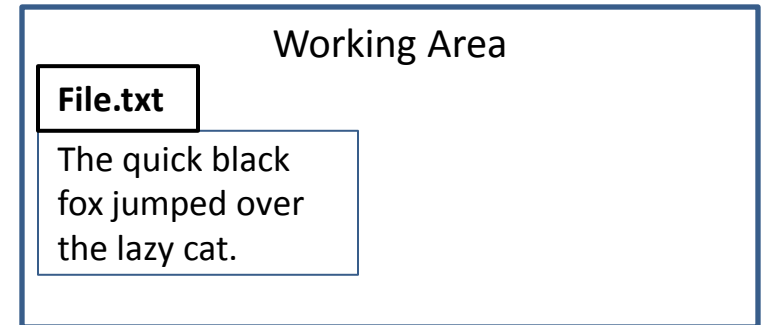
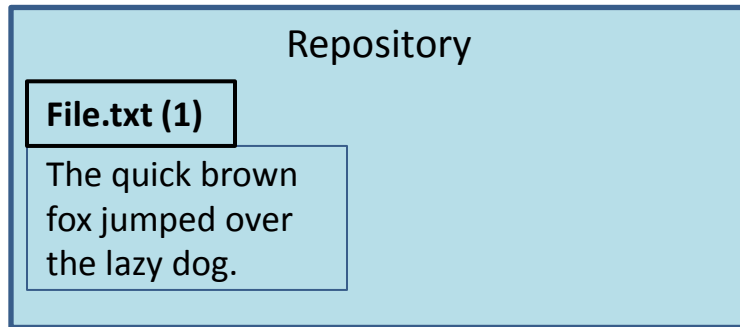
Check a Changed File into a Repository



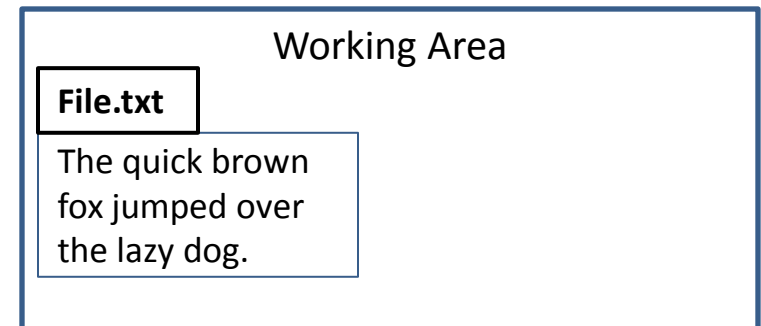
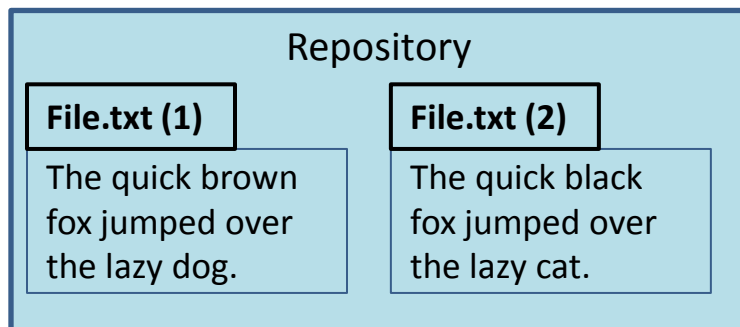
check File.txt into the Repository



Revert to a Previous Version in a Repository



check version 1 of File.txt out to the Work Area



A Potential Problem

We'll have multiple copies of File.txt that are almost the same –won't this waste space?

Yes – unless we use *deltas*.

- Usually changes from version to version are small.
- We can save space by only saving the changes (deltas).
- Basically, we need additions, deletions, changes.

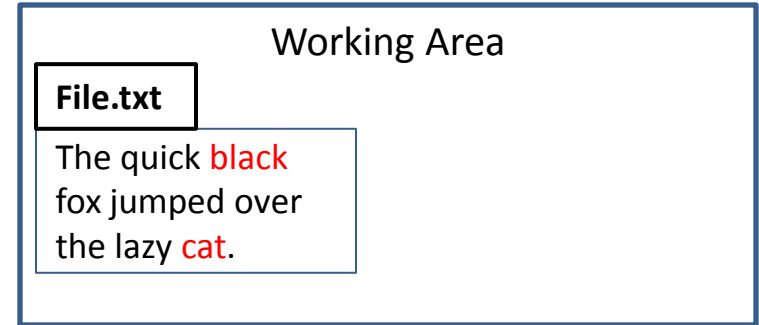
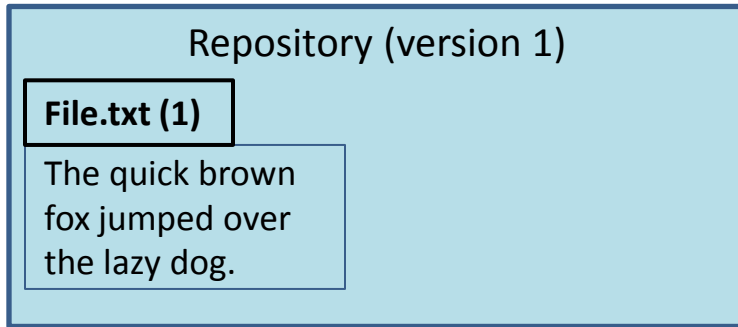
Example: **a** *Ino appended_text*

d *Ino*

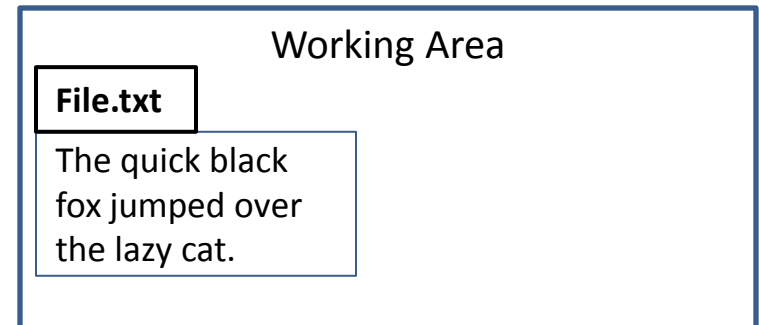
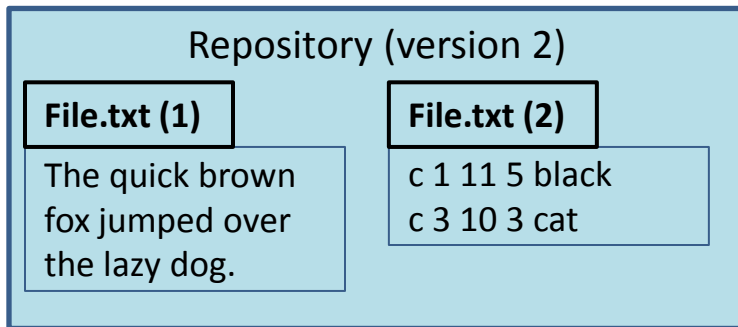
c *Ino start length new_text*

With smart differences and compression, deltas become very small.

Check in a Changed File (w/deltas)



check File.txt into the Repository (using deltas)



File vs. Repository Versioning

Versioning by file:

- Each file in repository has its own version number.
- Frequently changed files have higher numbers than stable files.
- May be difficult to find all the individual files representing one logical version.

Versioning by repository:

- Any changes update the version number of the entire repository.
- Easy to find all files comprising a given system version.
- Harder to find specific version of a given file.

Centralized vs. Distributed Repositories

Centralized:

- One master directory.
- All changes (by any team member) are applied to the master.
- Difficult for individuals to leave bread crumbs for their own experiments.

Distributed:

- Every developer has own repository.
- Changes are done to local repository.
- If working on a team, periodically PUSH local changes to designated central repository.

Sample of Version Control Systems

- **CVS** – Concurrent Versioning System
 - Centralized
 - File versioning
 - Used in CS3
- **git** – from Linus Torvalds, creator of Linux
 - Decentralized
 - Repository versioning
 - Used in this course
- Some others you may encounter
 - SVN – Subversion: Centralized, repository versioned
 - RCS – Revision Control System: Centralized, file versioned
 - SCCS – Source Code Control System: Centralized, file versioned