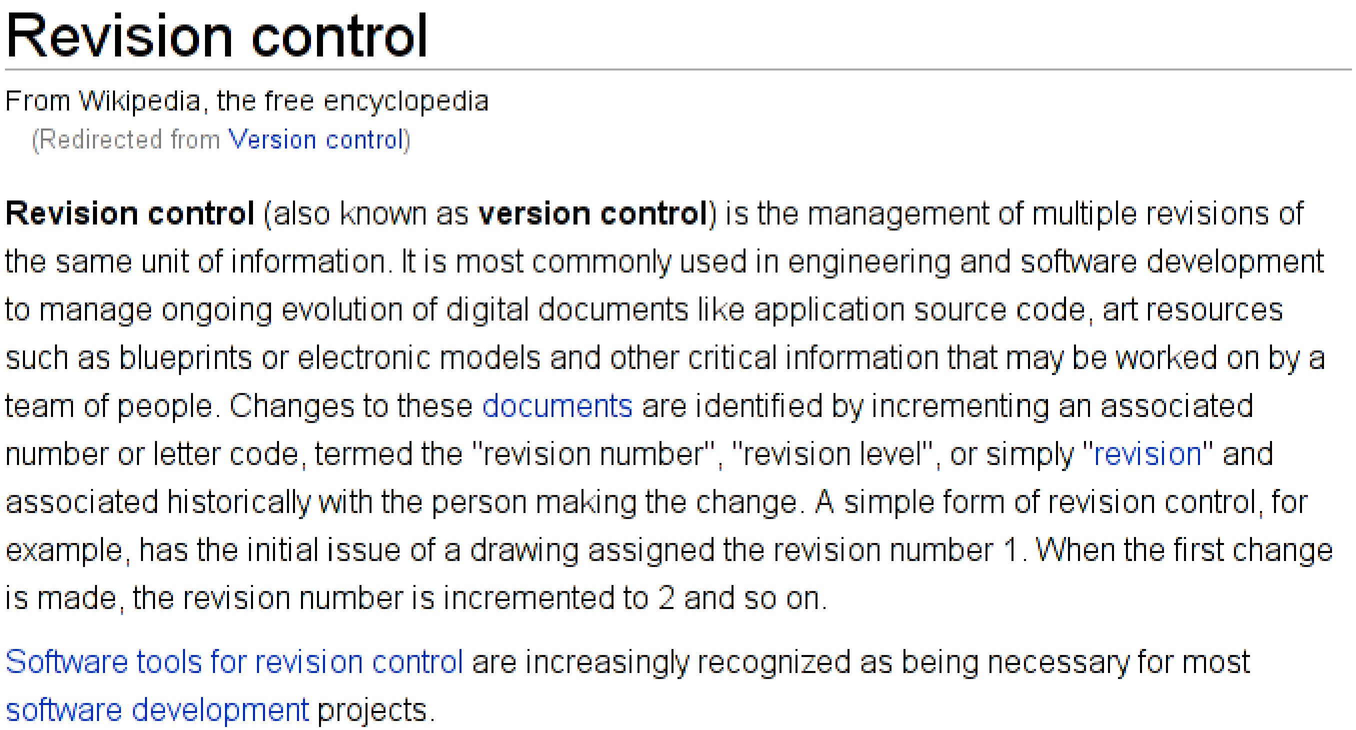
Version control with subversion

A short introduction

Outline

- What is version control?
- Aspects of Subversion
- Subversion basics

What is version control



Revision control

From Wikipedia, the free encyclopedia
(Redirected from Version control)

Revision control (also known as **version control**) is the management of multiple revisions of the same unit of information. It is most commonly used in engineering and software development to manage ongoing evolution of digital documents like application source code, art resources such as blueprints or electronic models and other critical information that may be worked on by a team of people. Changes to these documents are identified by incrementing an associated number or letter code, termed the "revision number", "revision level", or simply "revision" and associated historically with the person making the change. A simple form of revision control, for example, has the initial issue of a drawing assigned the revision number 1. When the first change is made, the revision number is incremented to 2 and so on.

Software tools for revision control are increasingly recognized as being necessary for most software development projects.

What is version control?

Version control is a methodology in which changes between different versions of your files are logged (and stored):

– Content	(what has changed?)
– Date	(when did it change?)
– Reason	(why has it changed?)
– Author	(who changed it?)

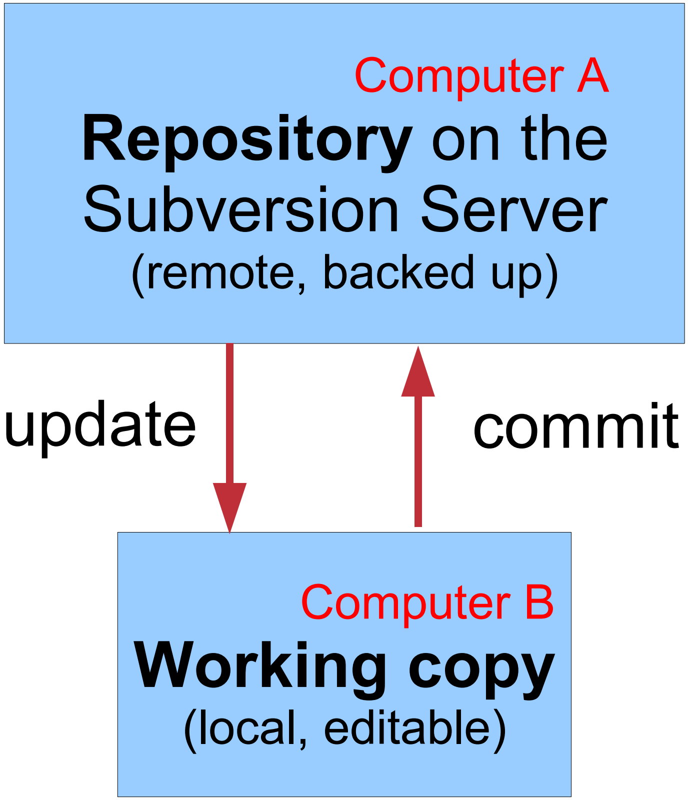
Advantages of version control

- Direct advantages:
  - File (and so project) history
  - Can recover older versions
  - Backup of project
- Indirect advantages
  - Dissemination
  - Collaboration between different authors
- For Text and Code:
  - Aids debugging (can compare versions)

Aspects of subversion

- Centralised version control system
- Developed as a successor to CVS
  - very similar to CVS
  - addresses the biggest shortcomings of CVS
- Web link: http://subversion.tigris.org
- Can communicate via http (WebDAV)
  - access to files is possible via a web browser
  - gives us very good control of users
    - who can access what?

Subversion basics



Computer A

Repository on the Subversion Server

(remote, backed up)

update

commit

Computer B

Working copy

(local, editable)

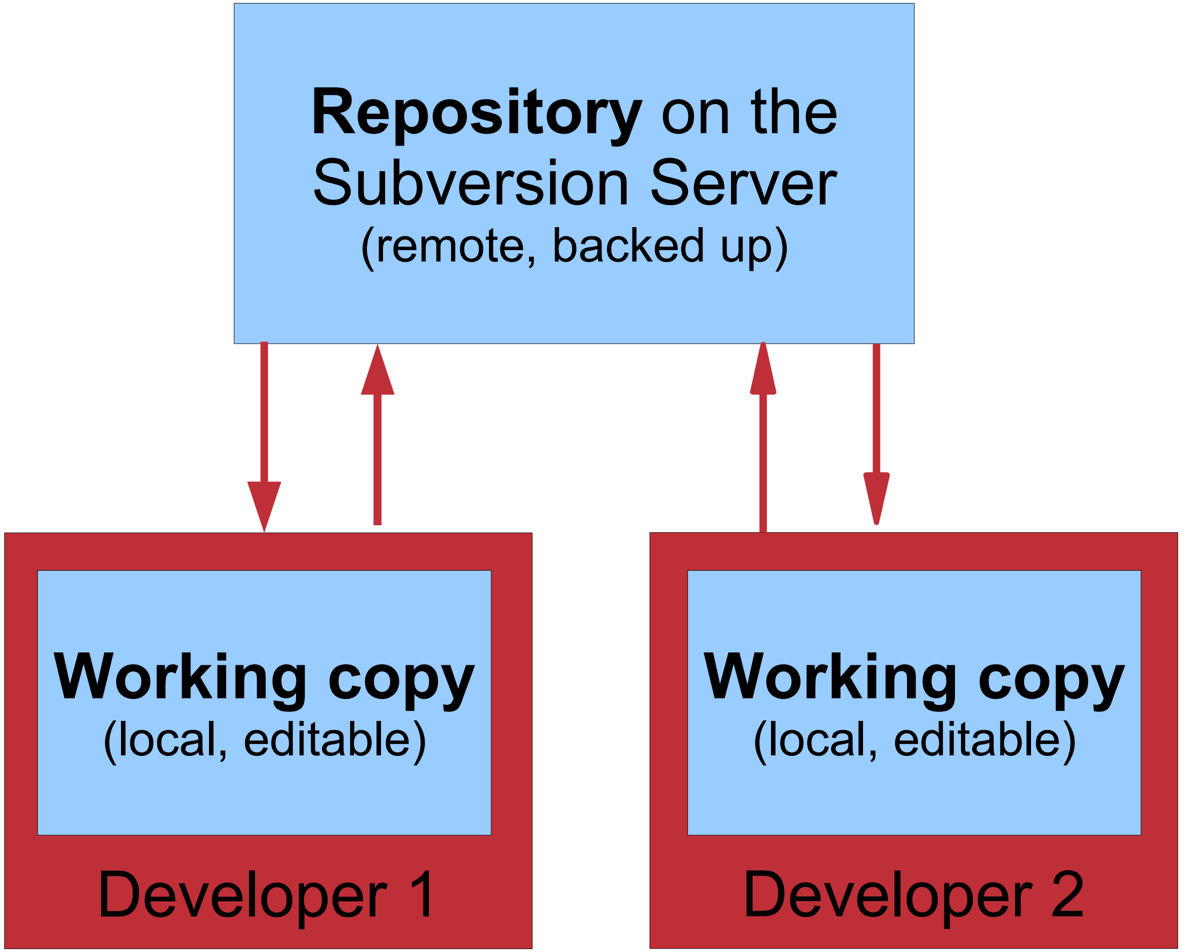
Subversion basics

- Day to day work:
  - Send your modifications: **commit**
  - Receive modifications by others: **update**
  - View history: **log**
- Setup of the working copy
  - **checkout**
- Setup of the repository
  - **import** (Note, it imports **into** the repository)

Subversion basics

- Operations on files/folders:
  - Add: **add**
  - Remove: **del**
  - Check changes: **diff**
  - View history: **log**
  - Rename: **move**
  - Undo: **revert**
- More advanced
  - Resolve conflicts: **resolved**
  - Apply changes between branches: **merge**

Subversion basics



Repository on the Subversion Server (remote, backed up)

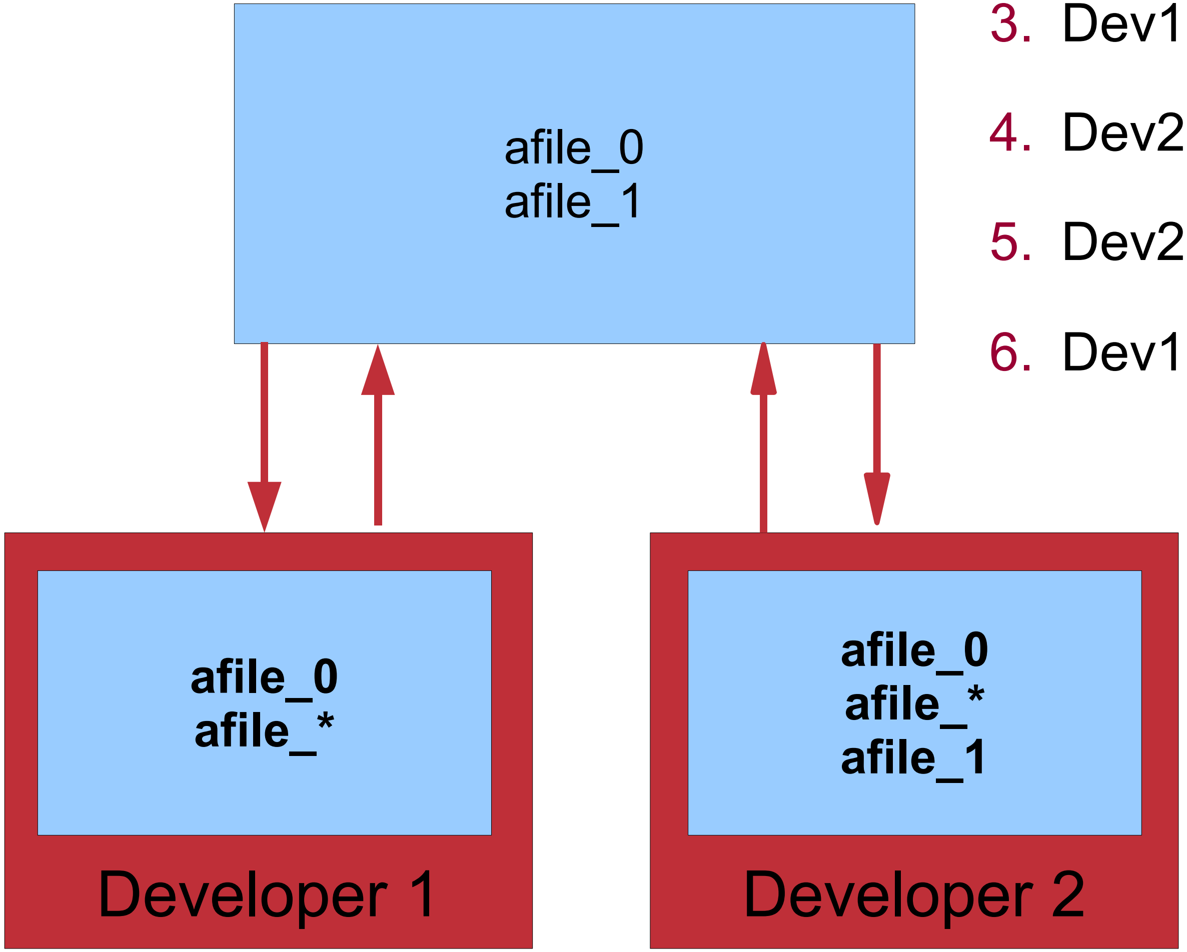
Working copy (local, editable)

Developer 1

Working copy (local, editable)

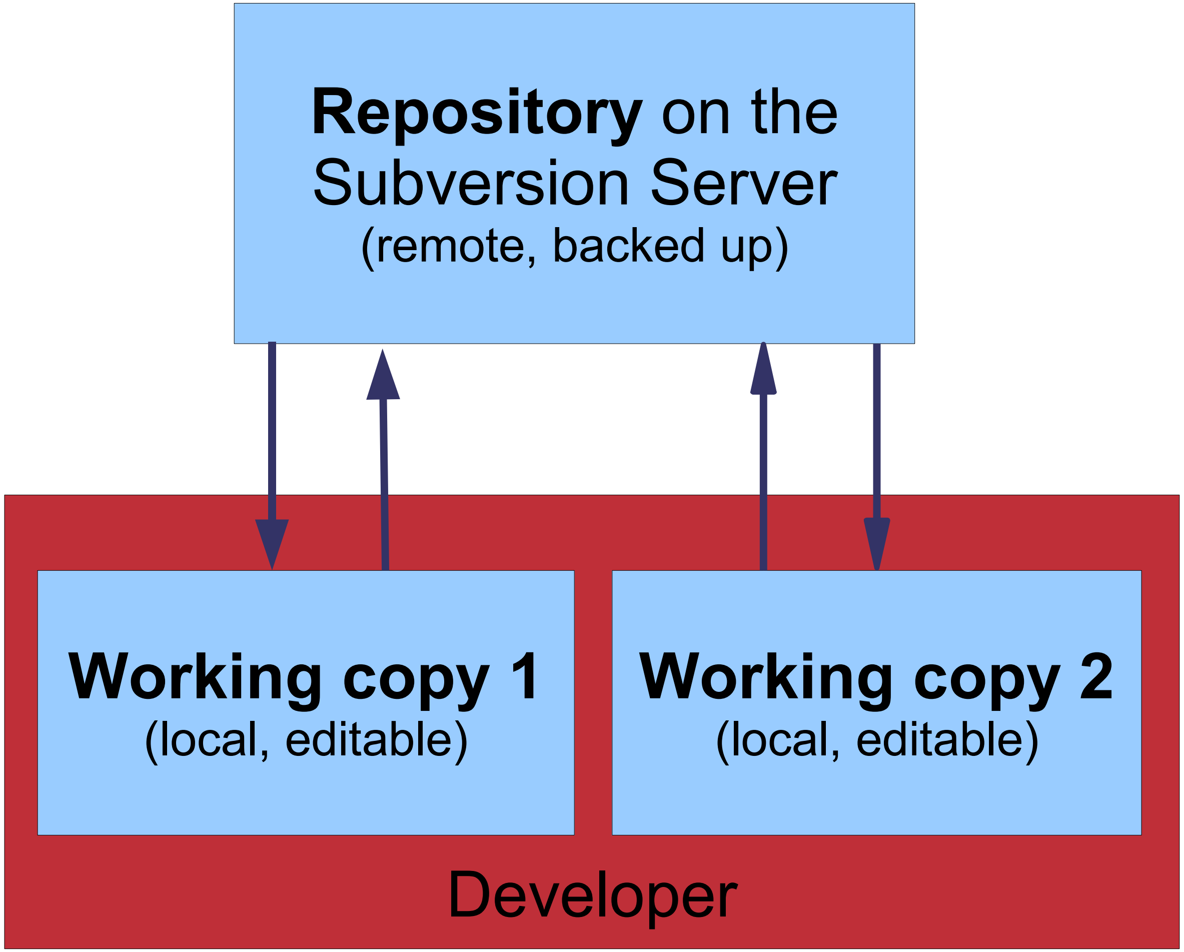
Developer 2

Potential conflicts



1. Dev1 checks out `afile_0`
2. Dev2 checks out `afile_0`
3. Dev1 modifies `afile`
4. Dev2 modifies `afile`
5. Dev2 commits `afile`
6. Dev1 commits / updates

Multiple working copies



Repository on the Subversion Server
(remote, backed up)

Working copy 1
(local, editable)

Working copy 2
(local, editable)

Developer