Branching (version control)

Branching, in version control and software configuration management, is the duplication of an object under version control (such as a source code file or a directory tree). Each object can thereafter be modified separately and in parallel so that the objects become different. In this context the objects are called **branches**. The users of the version control system can branch any branch.

Branches are also known as *trees*, *streams* or *codelines*. The originating branch is sometimes called the *parent branch*, the *upstream branch* (or simply *upstream*, especially if the branches are maintained by different organizations or individuals), or the *backing stream*. *Child branches* are branches that have a parent; a branch without a parent is referred to as the *trunk* or the *mainline*.[1]

In some distributed revision control systems, such as Darcs, there is no distinction made between repositories and branches; in these systems, fetching a copy of a repository is equivalent to branching.

Branching also generally implies the ability to later merge or *integrate* changes back onto the parent branch. Often the changes are merged back to the trunk, even if this is not the parent branch. A branch not intended to be merged (e.g. because it has been relicensed under an incompatible license by a third party, or it attempts to serve a different purpose) is usually called a *fork*.

Contents

Motivations for branching

Branches allow for parts of software to be developed in parallel.[2] Large projects require many roles to be filled, including developers, build managers, and quality assurance personnel. Further, multiple releases on different operating system platforms may have to be maintained. Branches allow contributors to isolate changes without destabilizing the codebase, for example, fixes for bugs, new features,[3] and versions integration. These changes may be later merged (resynchronized) after testing.

Development branch

A *development branch* or *development tree* of a piece of software is a version that is under development, and has not yet been officially released. In the open source community, the notion of release is typically metaphorical, since anyone can usually check out any desired version, whether it be in the development

branch or not. Often, the version that will eventually become the next *major* version is called *the* development branch. However, there is often more than one subsequent version of the software under development at a given time.

Some revision control systems have specific jargon for the main development branch; for example, in CVS, it is called the "MAIN"; in Git it is called the "master". A more generic term is " trunk".

Shadow or magic branches

In cvc and CVSNT, a *shadow* or *magic* branch "shadows" changes made in the upstream branch, to make it easier to maintain small changes (cvc is an open-source package building system incorporating a revision-control system for packages produced by rPath.)

Repository clones

In distributed revision control, the entire repository, with branches, may be copied and worked on further. Monotone (mtn), Mercurial (hg) and git call it "clone"; Bazaar calls it "branch".

See also

- Trunk (software)
- Revision tag

References

1. Berczuk, Steve; Appleton, Brad (2003). *Software Configuration Management Patterns: Effective Teamwork, Practical Integration* (http://www.scmpatterns.com/book). Addison-Wesley. ISBN 0-20174117-2. Retrieved 2007-05-24.
2. Appleton, Brad; Berczuk, Stephen; Cabrera, Ralph; Orenstein, Robert (1998-02-08). "Streamed Lines: Branching Patterns for Parallel Software Development" (http://www.hillside.net/plop/plop98/final_submissions/P37.pdf) (PDF). Hillside. Retrieved 2009-08-12.
3. Bailey, Derick (2009-07-15). "Part 1: Why" (http://www.lostechies.com/blogs/derickbailey/archive/2009/07/15/branch-per-feature-source-control-part-1-why.aspx). *Branch-Per-Feature Source Control*. Los techies. Retrieved 2009-08-12.