

Subversion

Version Compatibility



Here's some advice on choosing the right release of Subversion.

The toughest job for any server administrator is to choose the correct version of software to install and use for maintenance. Most server software have corresponding clients that access the software to get data from them. This kind of client-server model creates a few problems when a server administrator is trying to choose the right version of server software.

In this article, we will discuss one such problem in choosing the appropriate release of the popular version control system, Subversion. This article will also help decipher version compatibility among most of the free software available.

Servers

All the Subversion servers that have the same major version number will be compatible with each other. For example, if you are using Subversion 1.6.3 and want to switch to 1.6.x, which is greater than 1.6.3, both are compatible without any modifications. On the other hand, when you want to switch from 1.5.x to 1.6.x server, there may be some difference in the repository *backend.fsfs/bdb*, for which

the Subversion community provides tools to make this upgrade smooth without any problems. The Subversion servers that have a major version number will be compatible with all Subversion clients, which will have the same major version number.

Clients

When the minor version number (which is '6' in version 1.6.4) of a Subversion client changes, you'll typically notice more features added to the Subversion client, but without breaking the backward compatibility between the minor version numbers. In other words, none of the features available in the same minor version number series will be broken when we go for an upgrade. All the features in 1.5.x will continue to be available in 1.6.x along with some additional features.

The Subversion clients and servers will have sub minor version numbers such as the 4 in version 1.6.4. These sub minor versions are released with bug fixes. There is a good chance that these bug fixes are also back-ported to earlier releases, such as 1.5.x or 1.4.x, if these branches are still being maintained. But none of the features

introduced are backported to a previous minor version release. For example, tree conflicts introduced in 1.6.0 are not backported to the 1.5.0 release branch.

Let us take the case of the Subversion 1.6.x release and see how it gets affected between the bug fix releases. When Subversion 1.6 was officially released, it got the version number 1.6.0 with a number of new features, such as the following:

- Repository root relative URLs
- Detection of tree conflicts
- cTypes Python bindings
- Logging support for *svnserve*
- Support for file externals
- Sparse directory exclusion

Apart from the features introduced in this 1.6.0 release, there were many bug fixes and improvements to the existing features. Then, once in two months, Subversion developers started releasing 1.6.1, 1.6.2, etc. These included improvements and bug fixes on existing features and the features introduced in the 1.6.0 release.

There was a special case in the 1.6.x release branch, where the Subversion developers came up with a release 1.6.4, which was for just one security fix. In this release, bugs weren't fixed, but there was a single patch to fix a security issue in Subversion clients/servers. This could have been fixed as a part of a bug fix release in Subversion 1.6.4 but the developers did not want to wait for all the bug fixes scheduled for 1.6.4 to get in before they released the patch, which would also have to undergo a soak period of two months. The complete bug fix release that subsequently came out was version 1.6.5 .

In any release, if you upgrade to the same minor version—that is, 6 in 1.6.4 or 1.6.5—you will not experience any difference in the software. The dependencies that worked for 1.6.0 will continue to work for 1.6.4, 1.6.5, etc, unless you decide to change the dependencies to a different version due to some


security reasons.



Hence, a user can choose the following paths to upgrade to 1.6.6, which is the latest version, as of writing this article:

- 1.6.0 to 1.6.6
- 1.6.1 to 1.6.6
- 1.6.2 to 1.6.6
- 1.6.3 to 1.6.6
- 1.6.4 to 1.6.6
- 1.6.5 to 1.6.6

This also applies to changing from any arbitrary version to one in with the same minor number, like 1.6.2 to 1.6.4 and various similar combinations.

In case of Subversion 1.x.x and 2.x.x, all the above-mentioned compatibilities may or may not work. The Subversion community does not promise to preserve the backward compatibility between major versions of Subversion, in which case a 1.x.x client will not work smoothly with a 2.x.x server. Similarly, a 1.x.x server will not work smoothly with 2.x.x clients. Though we do not have a 2.x.x version of a Subversion release yet, this was the case with the 0.x.x Subversion release and the 1.x.x Subversion release.

 **Note:** Subversion 1.6.4 and 1.5.7 are important security fix releases and all servers must be updated for this fix, if Subversion servers using HTTP, HTTPS or the SVN protocols are exposed to the public.

The latest release of the Subversion (1.6.6) source can be downloaded from <http://subversion.tigris.org>. If you want Subversion binaries for different platforms, visit <http://open.collab.net>.  

By: Senthil Kumaran S.

The author is currently employed by CollabNet, and is working for the Version Control group. He is a 'full committer' of the Subversion project and a free software enthusiast. To know more, visit www.styleesen.org.