**Introduction**

VisualSVN is a transparent integration of the Subversion version control system to the Visual Studio development environment. VisualSVN allows you to take full control on any changes in the project that are made by you or your colleagues. With VisualSVN you can easily see the full history of modifications and restore previous versions of your project. So, you can treat it as a “smart time machine”.

VisualSVN is built on the base of the Subversion open source version control system that is a de-facto standard storage system for software projects. There are a lot of tools and services available for Subversion such as code review systems and hosting providers. With VisualSVN you can use these tools and services without limitations.

**What is Subversion?**

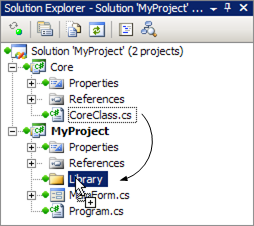
Subversion is a centralized multiuser version control system. Project files and full history of modifications are stored in a *repository*. The repository can be located both on a server, and on a local computer.

Each developer has his own copy of the project on the local computer, which is called the *working copy*. You can freely make any changes in your working copy, such as modifying files, adding new files, renaming files and folders and so on. And you don't even need a permanent connection to the repository. For example, you can work on your laptop on a plane. When a logical bundle of modifications is ready, you can upload it to the repository as a single atomic *commit*. Subversion automatically prevents overwrites of modifications that are made by another developer. Moreover, in most cases Subversion automatically merges modifications.

**What is VisualSVN?**

Subversion is a perfect version control system. VisualSVN makes Subversion easy to use for Visual Studio developers. By integrating Subversion to Visual Studio VisualSVN improves your productivity and reduces the probability of routine mistakes.

VisualSVN is conceptually based on the principle of “transparent Subversion”. VisualSVN automatically reflects all actions to Subversion from Visual Studio that happen in the daily development cycles. For example, VisualSVN automatically marks all newly added files and folders as *added*. This transparent behavior allows you to concentrate on development, while VisualSVN takes care off versioning.

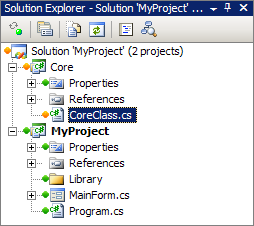


Moreover, VisualSVN allows you to manage files in the Solution Explorer in a transparent fashion. For example, you can drag-and-drop files between projects in the Solution Explorer and VisualSVN transparently reflects this operation to Subversion, with full history preservation. VisualSVN supports the complete list of file management operations including addition, deletion, copying, renaming and drag-and-drop. Transparent file management allows you to refactor without pain.

**Intuitive status**

VisualSVN displays *traffic lights* status for almost all items in the Solution Explorer. There are only three statuses:

* green - there are no changes,
* yellow - there are some changes,
* red - there are conflicts or errors.



For each compound item such as folder or project VisualSVN displays the cumulative status. For example, there are some changes within a project's folder if the status of the corresponding item in the Solution Explorer is yellow. For the topmost solution item in the Solution Explorer VisualSVN displays the cumulative status of the entire working copy. For convenience, VisualSVN repeats the cumulative status of the entire working copy in the right corner of Visual Studio status bar.

**Unlimited Subversion**

In addition to such unique capabilities as transparent file management and intuitive status display, VisualSVN provides convenient access to all Subversion's commands using the mature and stable graphical user interface of TortoiseSVN, the de-facto standard Subversion client for the Windows platform. Thanks to this fact, you get unified access to Subversion both in Visual Studio, and Windows Explorer.



VisualSVN is a professional Subversion integration to the Visual Studio development environment. The functionality of VisualSVN completely covers the needs of the software development process including initial adding of a project under Subversion, refactoring, branching and merging.