



SourceGear Vault vs. Subversion

A question we hear frequently, and understandably so, is: "Why should we choose Vault when Subversion is available for free?" This paper explains, in brief, the reasons why many of our customers have made precisely that choice.

In short – Vault's powerful features save developers' time, cutting costs and getting projects shipped sooner. And SourceGear's dedicated support team brings our customers up to speed sooner, and keeps things running smoothly – again, saving time, saving money. Our customers believe Vault is a bargain when all is said and done.

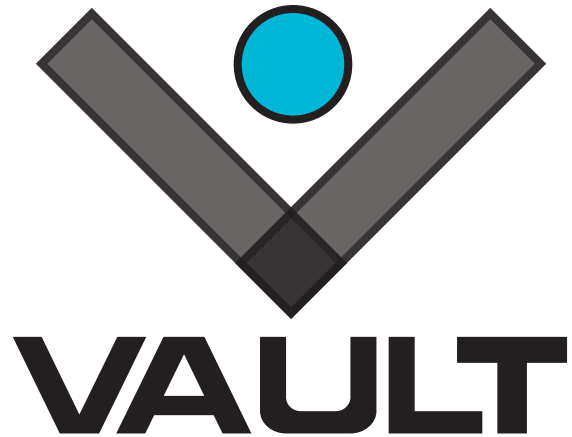
Features

Subversion and Vault solve much the same basic problem – both are modern, fast version control tools, offering cheap branches, concurrent development, and more. Both are huge improvements over older systems such as CVS and SourceSafe – with atomic commits, and edit/merge/commit development instead of troublesome exclusive locks.

Vault does, however, offer features not available in Subversion (and in some cases, not available anywhere else).

Line History

Vault's Line History feature is a SourceGear exclusive. Where Subversion (and others, including Vault) offer "blame" as a way to see what developer last changed a particular line of code, Line History takes the "what happened here" concept much further. Line History allows the programmer to select any chunk of a source file, and go back in time to the last checkin that touched those lines – with checkin comments and diffs a click away. If that change isn't the one we're interested in, we can go back to the previous checkin affecting those lines... and so on. Line History skips over in-between revisions, and we find exactly the history we need, in seconds.



The Version Control Tool
For Small and Midsize Teams

May, 2010

Merge Branches Wizard

Subversion and Vault, like most modern version control systems, make it easy to create any number of parallel branches of code. Branches might represent the current production release of a project vs. the next-generation development code. Further branches may be created as needed for production bug fixes, etc.

But if a bug is fixed on a production branch, how do we migrate that fix to the development code?

In Subversion, it's certainly doable, but a completely manual process. You need to know ahead of time:

1. The branch from which you're moving changes
2. The branch to which you're migrating the new code
3. The version of the source branch from which to pull the changes
4. If you're only moving a subset of the branched code, you need to get the source and destination paths just right.

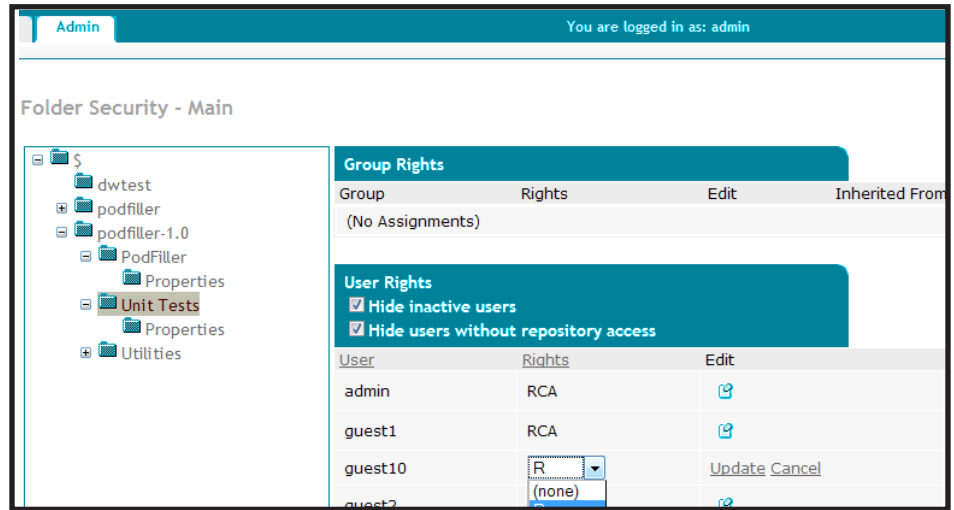
This is all typically done from the command line. And the merges can't be narrowed to a specific checked-in set of files without specifying them individually.

In Vault, the Merge Branches Wizard lets you point out the source and destination branches. It then offers a list of change sets, including descriptions, to merge. 20 different files involved in the "avoid crashing on invalid phone numbers" fix? No problem, Vault will merge all of them, and only those. No missed changes, no code moved too early.

Vault can also merge a wider range of changes – such as file and directory renames – than Subversion.

Folder Security

Subversion source code database access is an all-or-nothing proposition. A developer who can



Line History

see part of the repository can see the whole thing.

This is less-than-ideal when different teams need different access levels. Vault offers folder-level security, by user or by group. This allows teams to:

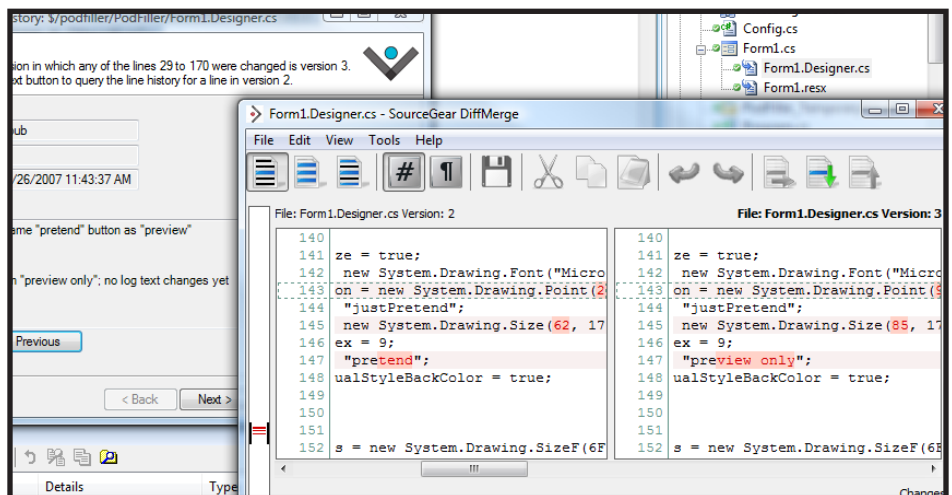
- Share library code as a read-only repository to all teams
- Hide entire sections of the repository from contractors, partners, etc. while allowing complete access to the core team
- "Lock down" production, QA, or deployment branches so that only certain users can allow code to move to production, etc.

SourceSafe Import

Vault was designed specifically as a VSS replacement. If you need to import old source code from VSS, for example, our import tools will be extremely useful. We offer a much easier and more reliable upgrade of your old data, and a more-familiar experience as well. Most of our customers who upgrade from VSS tell us there's virtually no learning curve with Vault.

SourceSafe Feature Support

For teams moving away from VSS, Vault is a life-saver. Apart from all the modern, reliable version control features Vault brings, we support SourceSafe-specific features, so no capabilities are lost in the transition.



Folder Security

Unlike Vault, Subversion does not support:

- Sharing files or folders. This allows automatically-synched copies of files to exist in multiple projects at once. So changing one file can cause, for example, a copyright notice to be updated in all Vault-controlled web sites.
- Pinning. A shared file may need to remain “stuck” at a particular revision in, say, a historical release branch. Pinning handles this.
- Shadow Folders. An extremely handy feature — when a Shadow folder (on the server, elsewhere on the network, etc.) is attached to a Vault folder, any checkins to that Vault project will automatically be updated to the shadow folder. So any checkins to a web site’s QA branch might be immediately available on the QA web server.

Integration

Version Control is not an end in itself, and is useless in isolation. Code needs to be created and updated by developers, in the environment they choose.

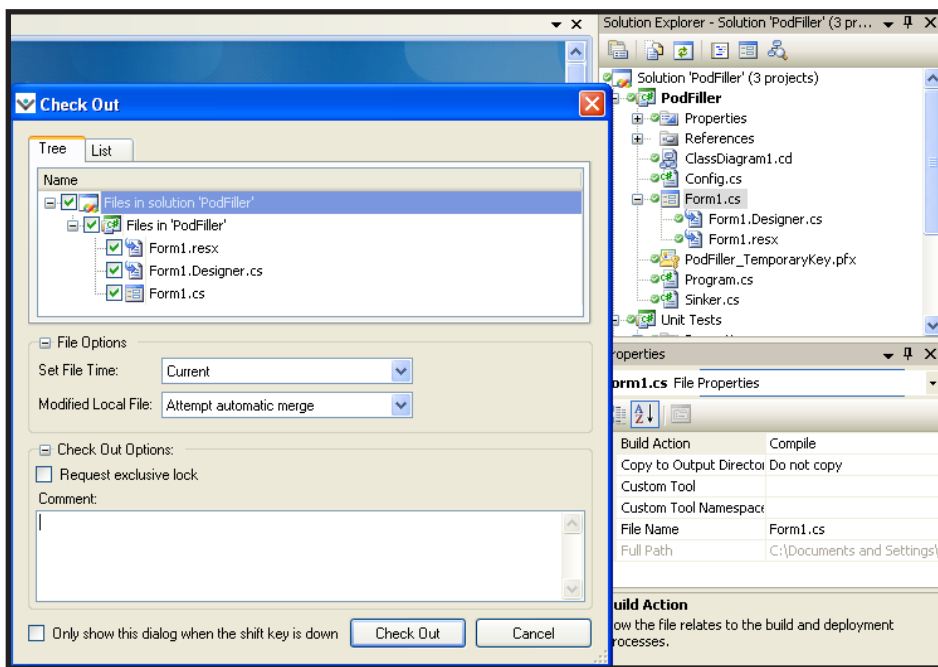
Vault’s standalone client is a usable, comfortable, powerful tool – but there’s nothing like having your version control work seamlessly with your development environment, fading into the background and “just working”. Vault’s IDE integration shines here.

While various integration packages are available to combine Subversion with editors and IDEs, none of them are part of Subversion itself, or even developed by the same team. So after picking one of the competing solutions for a particular environment, the developer will end up as a troubleshooting middleman when things go wrong.

Visual Studio Integration

Vault’s Visual Studio integration is a major point of pride. All of Vault’s power is here, and most of it completely effortless. Edit a file, and Vault notes immediately that there are changes pending. Getting the latest code, merging in others’ changes, and checking code in are a click away. Vault’s Pending Changes panel makes the current state of all project files clear. And SourceGear’s award-winning DiffMerge file comparison tool can be launched at any time to see exactly what’s changed.

Vault works with all Visual Studio .NET versions, as well as the older Visual Studio 6 environment.



Visual Studio Integration

Eclipse Integration

Not everyone uses Visual Studio – millions of developers use the Eclipse editor for development in almost any language, on many platforms.

Vault’s Eclipse integration offers all of the power of our Visual Studio client, and runs under Windows, Mac and Linux. Like all of Vault, it is fully-supported, and receives constant testing, updates and improvements.

Dreamweaver

Vault ships with a fully-supported Dreamweaver client (for Windows), so the web design team can harness the same power as the developers.

VSS-Compatible Integration

Vault’s “classic” SourceSafe emulation lets Vault work with many IDEs and editors supporting SourceSafe version control.

Architecture / IT

Vault’s server is Windows-based, written from the ground up using Microsoft .NET technology, which allows it to be deployed to a wide range of Windows systems. From

Windows 2000 through Vista Ultimate 64-bit, Vault is there.

While Subversion can be hosted on Windows, it’s not that project’s primary architecture, and it’s generally considered an option for the brave.

Installing Vault on a Windows server takes – literally – minutes. And Vault can optionally work with existing Windows domain accounts for all of

its login, folder security, and logging functions.

Ongoing maintenance of Vault is practically a non-issue; backups can be scheduled via many off-the-shelf tools, including simple SQL server scripts.

For a Windows shop, not needing a Linux guru on hand, or someone to debug Windows-hosted Subversion, is a huge reduction in stress and budget.

Support

If we were to ask our existing customers the main reason they choose Vault over free tools (and we have), they would answer “support” in great numbers (and they have).

Subversion has a number of support sites – bulletin boards, IRC, developer and user mailing lists. Google is the primary interface to all of these – there’s a chance that someone else has had the same problem, and even a chance that they’ve found, and posted, a solution. But the more-site-specific, tricky and confounding an issue, the less likely a matching case will be found.

At SourceGear, a dedicated support staff specializes in sorting through our customer’s issues quickly. They know the products inside-out, and their feedback helps us make each new release more reliable and powerful than the last. SourceGear support is available via email to support@sourcegear.com, or by calling us at (217)356-0105.

SourceGear has support forums, too – in one location, read by our support staff, developers, and product managers.

Whether it’s importing a particularly quirky SourceSafe database; helping get Vault running within a new Eclipse-based IDE; or helping a customer set up a branching strategy that works best for them; SourceGear support has been through it all.

And did we mention that Vault support is free?

Fortress

One more point worth mentioning: Vault is not SourceGear’s only product – for teams looking for version control and bug tracking, Vault’s big brother **SourceGear Fortress** is a slam-dunk. Fortress includes all of Vault’s features, seamlessly integrated with bug/feature/task tracking – all available in Visual Studio, Eclipse, our standalone client, or via the web.

