Gemini -TortoiseSVN Integration

By Robert B: Denny <rdenny@dc3.com

Version 1.5 12-Feb-2010 (for Gemini 3.6)

See the Copyright Notice at the end of this document

This package implements client-side SVN hooks for TortoiseSVN (1.6.6 or later) which allow it to publish commits to Gemini. It is similar in principle to the CounterSoft_Gemini_SourceControl_v3_5_4.zip package but it runs on the client computer (driven by TortoiseSVN) instead of at the SVN server. This makes it possible to integrate with Gemini regardless of the SVN server in use or the connection methodology to the SVN server. It has been tested with svn+ssh link to a remote Linux/svnserve/sshd server, but it doesn't matter what client-server protocol is used with TortoiseSVN and its SVN server.

Requirements

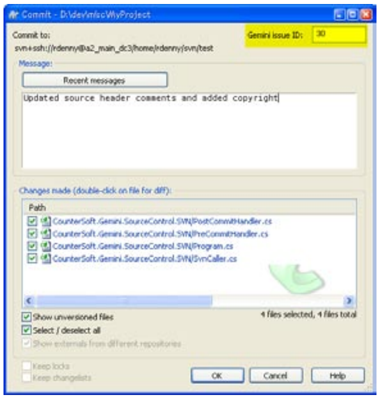
This version must be used with Gemini 3.6.

What it Does

A new field appears in the TortoiseSVN Commit window in which you enter an applicable Gemini Issue ID for the commit.

Before committing it checks the supplied Gemini Issue ID and if it is unknown to Gemini, an error message is displayed and the commit is not done. Checks are performed on any additional Gemini Issue IDs found in the commit message as well. In addition, if a Gemini Issue ID is supplied, but the commit message is empty, an error is displayed and the commit is not done.

After committing, a comment is added to the Gemini Issue, describing the files that were committed, when and by whom, the new SVN revision number, and the repository path. In addition, Source Control File Links are added to the Issue for affected files. For details see Using the Integration.



Commit - D:\dev\misc\MyProject

Commit to:
svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test

Gemini issue ID: 30

Message:

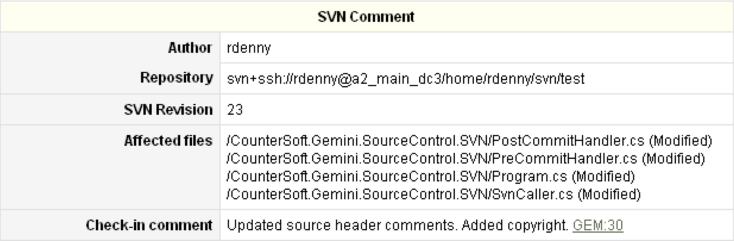
Updated source header comments and added copyright

Changes made (double-click on file for diff):

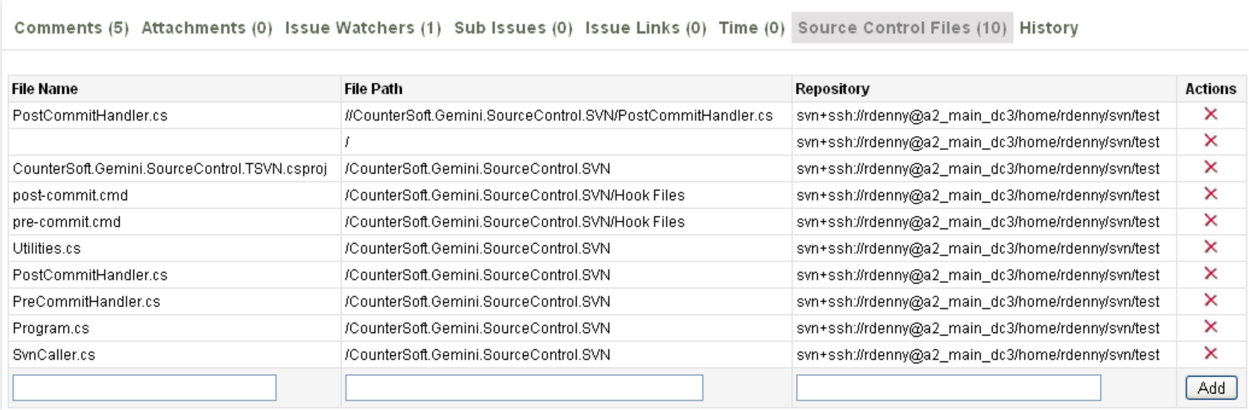
Path
- ☑ CounterSoft.Gemini.SourceControl.SVN/PostCommitHandler.cs
- ☑ CounterSoft.Gemini.SourceControl.SVN/PreCommitHandler.cs
- ☑ CounterSoft.Gemini.SourceControl.SVN/Program.cs
- ☑ CounterSoft.Gemini.SourceControl.SVN/SvnCaller.cs

4 files selected, 4 files total

☑ Show unversioned files
☑ Select / deselect all
☑ Show externals from different repositories
□ Keep locks
□ Keep changelists



SVN Comment	
Author	rdenny
Repository	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test
SVN Revision	23
Affected files	/CounterSoft.Gemini.SourceControl.SVN/PostCommitHandler.cs (Modified) /CounterSoft.Gemini.SourceControl.SVN/PreCommitHandler.cs (Modified) /CounterSoft.Gemini.SourceControl.SVN/Program.cs (Modified) /CounterSoft.Gemini.SourceControl.SVN/SvnCaller.cs (Modified)
Check-in comment	Updated source header comments. Added copyright. GEM:30



File Name	File Path	Repository	Actions
PostCommitHandler.cs	//CounterSoft.Gemini.SourceControl.SVN/PostCommitHandler.cs	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
	/	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
CounterSoft.Gemini.SourceControl.TSVN.csproj	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
post-commit.cmd	/CounterSoft.Gemini.SourceControl.SVN/Hook Files	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
pre-commit.cmd	/CounterSoft.Gemini.SourceControl.SVN/Hook Files	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
Utilities.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
PostCommitHandler.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
PreCommitHandler.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
Program.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
SvnCaller.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
			Add

Configure the Integration

Ensure that Gemini REST API is enabled on your web/SQL server. See the Gemini Management Guide for instructions. Edit the file *CounterSoft.Gemini.SourceControl.TSVN.exe.config* with Notepad or other plain text editor. Normally this is located in C:\Program Files\TortoiseSVN\Hooks. Change the following variables *carefully* to suit your installation:

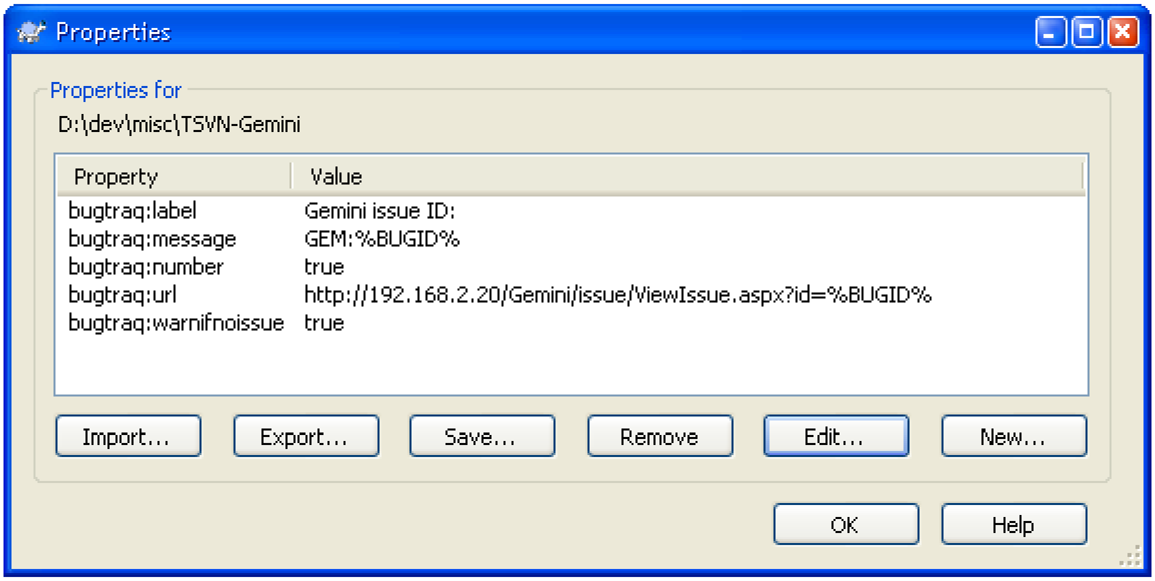
- *GeminiUsername* (and possibly UseWindowsAuthentication)
- *GeminiPassword*
- *GeminiURL* (URL of your Gemini system)

The other configuration variables are covered in the Advanced Configuration section.

Set Up a Project

Now it's time to set up a project for which you want to integrate SVN with Gemini.

1. Set the TortoiseSVN Properties for the root of the project exactly as shown in the figures below. The properties should be **applied recursively** to all of the files below the root. The file *gemini_svn.svnprops* in the TortoiseSVN Hooks folder may be imported to set these properties in one step, however you will still have to go back and edit each to make it recursive. **NOTE: Change the IP address or domain name in the bugtraq:url property to the that of your Gemini system.**

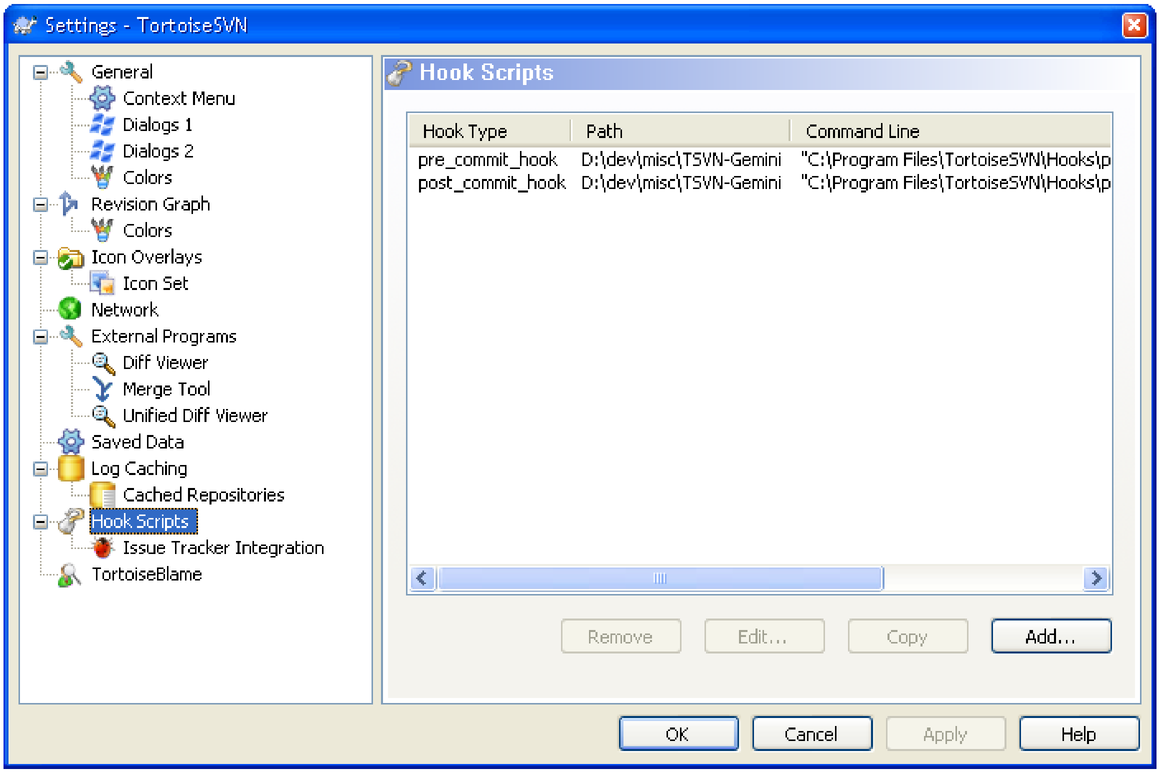


Properties for
D:\dev\misc\TSVN-Gemini

Property	Value
bugtraq:label	Gemini issue ID:
bugtraq:message	GEM:%BUGID%
bugtraq:number	true
bugtraq:url	http://192.168.2.20/Gemini/issue/ViewIssue.aspx?id=%BUGID%
bugtraq:warnifnoissue	true

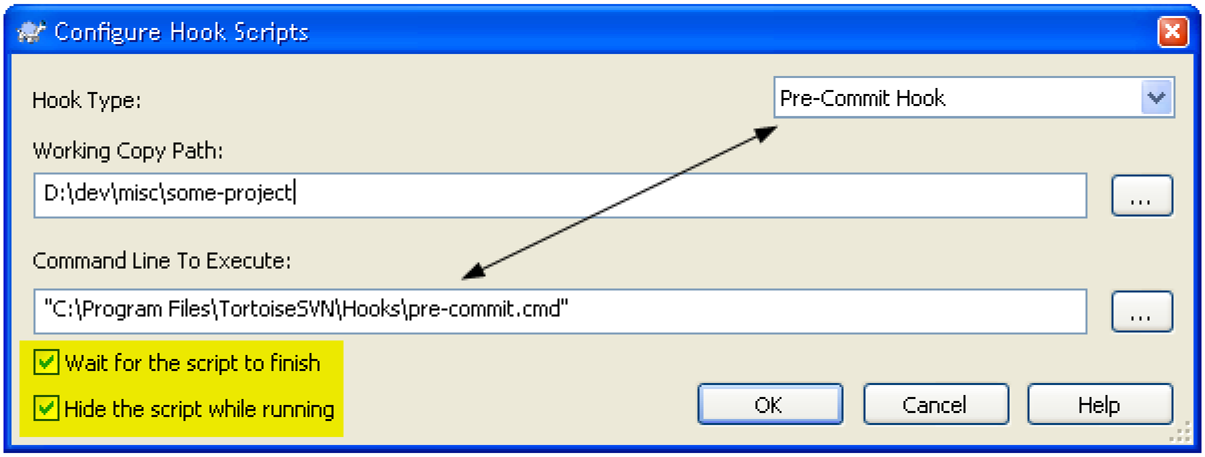
These properties affect TortoiseSVN's user interface. Here's a quick explanation of them.

2. Next, inform TortoiseSVN that it should call into the integration hooks during its pre-commit and post-commit phases whenever it's asked to work on the project. Tortoise needs to know when it is looking at a file within the project, and if so, call the hooks. This is done in TortoiseSVN's Settings editor, in the Hook Scripts section:



Hook Type	Path	Command Line
pre_commit_hook	D:\dev\misc\TSVN-Gemini	"C:\Program Files\TortoiseSVN\Hooks\p
post_commit_hook	D:\dev\misc\TSVN-Gemini	"C:\Program Files\TortoiseSVN\Hooks\p

Make two entries, pre-commit and post-commit, for your project's root folder. In the image above, the Path is the physical disk path to the root of the project. The Command Line is the path and file name of one of the .cmd files in the Tortoise *hooks* folder (see the Installation section). It looks like this in the Configure Hook Scripts window:



Hook Type: Pre-Commit Hook

Working Copy Path: D:\dev\misc\some-project

Command Line To Execute: "C:\Program Files\TortoiseSVN\Hooks\pre-commit.cmd"

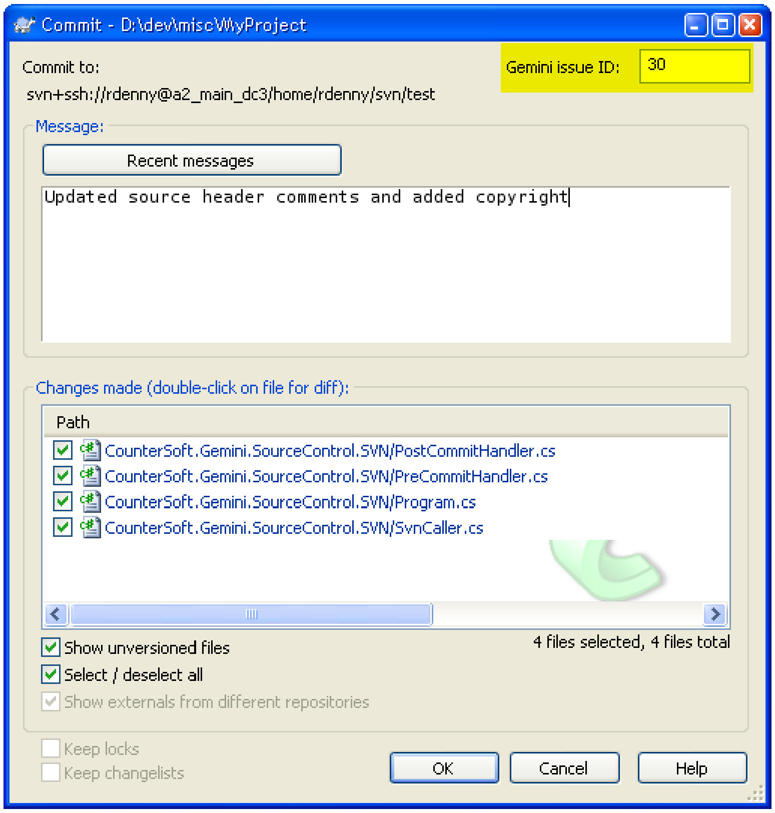
☑ Wait for the script to finish

☑ Hide the script while running

The Command Line to Execute should specify the appropriate .cmd file for pre-commit or post-commit. The Working Copy Path is the project root. This will tell TortoiseSVN that it should use the Gemini Integration hooks for all commits at or below this root. Finally, note the highlighted checkboxes. Be sure to turn both of these on, or you'll see ugly console windows pop up and the pre-commit hook won't work to prevent committing with non-existent Gemini Bug IDs.

Using the Integration

When you commit files or folders within a project that you have set up as described above, you'll see a new field Gemini Issue ID in the upper right corner of the TortoiseSVN Commit dialog. Enter the Gemini issue ID to which the commit applied in this field. Then do the commit. Once it completes, you will discover that a new comment has been added to the Gemini issue, describing the files added, changed, etc., who did the commit and when, the new SVN revision number, and the repository path. In addition, Source Control File links for the affected files are added to the issue in Gemini. Here's what a commit looks like:



Commit - D:\dev\misc\MyProject

Commit to:
svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test

Gemini issue ID: 30

Message:

```
Updated source header comments and added copyright
```

Changes made (double-click on file for diff):

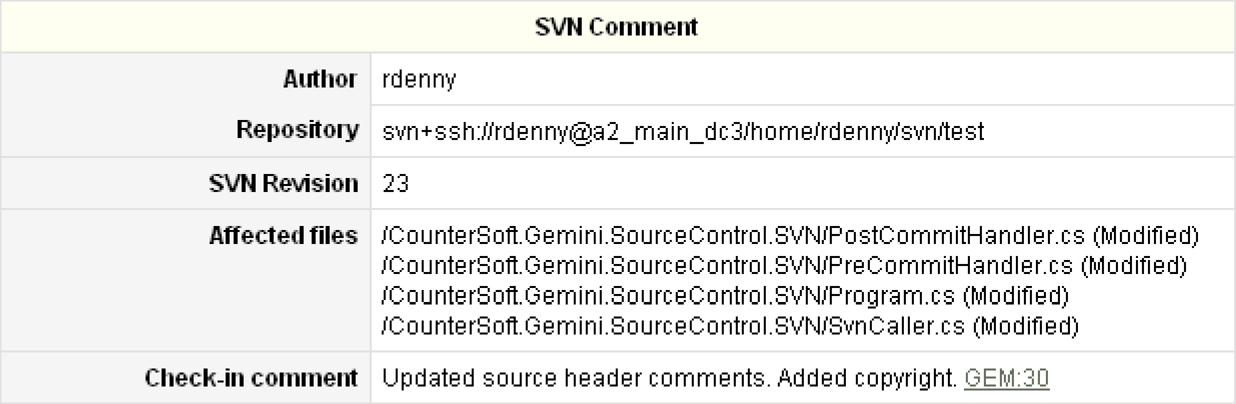
Path
☑ CounterSoft.Gemini.SourceControl.SVN/PostCommitHandler.cs
☑ CounterSoft.Gemini.SourceControl.SVN/PreCommitHandler.cs
☑ CounterSoft.Gemini.SourceControl.SVN/Program.cs
☑ CounterSoft.Gemini.SourceControl.SVN/SvnCaller.cs

4 files selected, 4 files total

☑ Show unversioned files
☑ Select / deselect all
☑ Show externals from different repositories

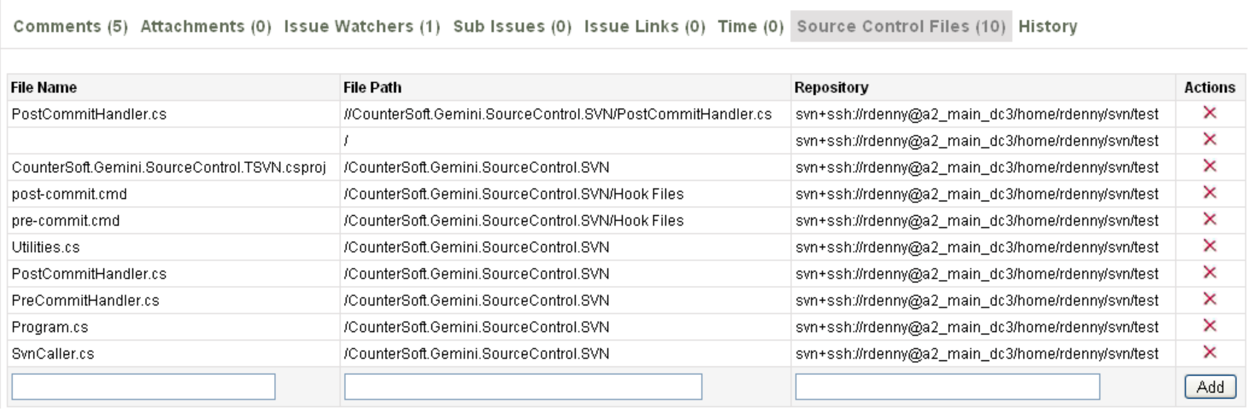
□ Keep locks
□ Keep changelists

and here is what the corresponding comment in Gemini looks like:



SVN Comment	
Author	rdenny
Repository	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test
SVN Revision	23
Affected files	/CounterSoft.Gemini.SourceControl.SVN/PostCommitHandler.cs (Modified) /CounterSoft.Gemini.SourceControl.SVN/PreCommitHandler.cs (Modified) /CounterSoft.Gemini.SourceControl.SVN/Program.cs (Modified) /CounterSoft.Gemini.SourceControl.SVN/SvnCaller.cs (Modified)
Check-in comment	Updated source header comments. Added copyright. GEM:30

and here is what the Source Control Files info looks like:



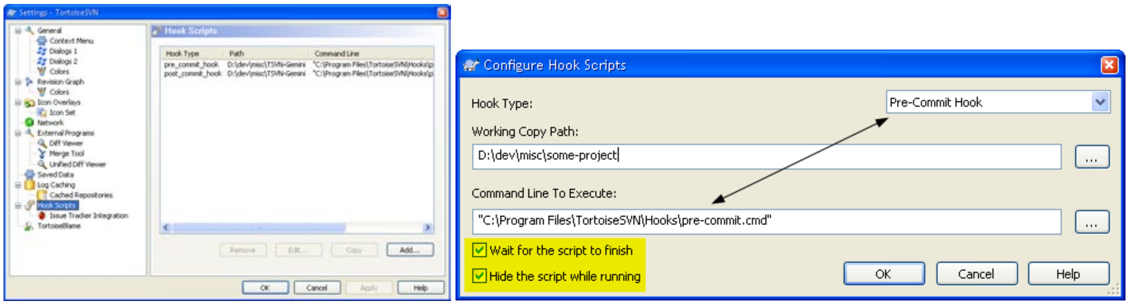
File Name	File Path	Repository	Actions
PostCommitHandler.cs	//CounterSoft.Gemini.SourceControl.SVN/PostCommitHandler.cs	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
	/	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
CounterSoft.Gemini.SourceControl.TSVN.csproj	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
post-commit.cmd	/CounterSoft.Gemini.SourceControl.SVN/Hook Files	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
pre-commit.cmd	/CounterSoft.Gemini.SourceControl.SVN/Hook Files	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
Utilities.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
PostCommitHandler.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
PreCommitHandler.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
Program.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
SvnCaller.cs	/CounterSoft.Gemini.SourceControl.SVN	svn+ssh://rdenny@a2_main_dc3/home/rdenny/svn/test	✕
			Add

Gemini Issue IDs in SVN commit messages

When you put a Gemini Issue ID into the TortoiseSVN Commit dialog, this causes the string "GEM:*nn*" to be added at the end of the commit message (where *nn* is the Gemini Issue ID). If you want to cross-reference additional Gemini Issues, you can manually enter additional Gemini Issue IDs as GEM:*nn* in the body of the commit message. The integration hook will pick them all up and add commit comments to all referenced Gemini Issues.

Troubleshooting

The first thing to look at if your commits are not being recorded in Gemini is the Hook Scripts settings for the root of your project. See Set Up a Project above.



Hook Type	Path	Command Line
pre_commit_hook	D:\dev\misc\TSVN-Gemini	"C:\Program Files\TortoiseSVN\Hooks\p
post_commit_hook	D:\dev\misc\TSVN-Gemini	"C:\Program Files\TortoiseSVN\Hooks\p

Hook Type: Pre-Commit Hook

Working Copy Path: D:\dev\misc\some-project

Command Line To Execute: "C:\Program Files\TortoiseSVN\Hooks\pre-commit.cmd"

☑ Wait for the script to finish

☑ Hide the script while running

Usually, you'll find that there aren't Hook Script entries for the root of your project (or a Path is mistyped).

If you don't see the Gemini Issue ID in the upper right corner of the TortoiseSVN Commit dialog, you haven't added the *bugtraq* properties, or they weren't *applied recursively* from the root of your project down. See Set Up a Project above.

Is the .NET Framework 3.5 Service Pack 1 (or later) installed?

If the above are correct, then you'll have to dig deeper. The integration program sends debug tracing messages that can be useful. You can display them with the SysInternals DebugView utility. Install and run DebugView, then do a commit and see what is displayed. Usually, this will give you some hints as to what is not working.

Advanced Configuration

There are additional things which you can adjust in Gemini-TortoiseSVN Integration. Configuration files are located in the folder in which the hook components are installed. This is typically C:\Program Files\TortoiseSVN\Hooks\.

Disable Pre-Commit Issue ID Checking

You may find that committing takes a long time as issue IDs are checked live with Gemini. To completely disable this, edit the file *CounterSoft.Gemini.SourceControl.TSVN.exe.config* with Notepad or other plain text editor, and change the value of *PerformPreCommitChecks* from True to False.

Disable Addition of Issue Comments

By default, the addition of issue comments to Gemini for commits is enabled. You can turn this feature off. To disable this, edit the file *CounterSoft.Gemini.SourceControl.TSVN.exe.config* with Notepad or other plain text editor, and change the value of *AddIssueComment* from True to False.

Disable Addition of Source Control File Links

By default, the addition of issue Source Control File links to Gemini for commits is enabled. You can turn this feature off. To disable this, edit the file *CounterSoft.Gemini.SourceControl.TSVN.exe.config* with Notepad or other plain text editor, and change the value of *AddIssueSourceCodeFileLink* from True to False.

Force an Issue ID to be Supplied

You can force a user to supply an issue ID on every commit to a Gemini-enabled project tree. To enable this, edit the file *CounterSoft.Gemini.SourceControl.TSVN.exe.config* with Notepad or other plain text editor, and change the value of *RequireIssueID* from False to True.

Disable Requiring a Comment if an Issue ID is Supplied

Normally, if an issue ID is supplied on a commit, it is also required that a comment be supplied. To disable this, edit the file *CounterSoft.Gemini.SourceControl.TSVN.exe.config* with Notepad or other plain text editor, and change the value of *RequireComment* from True to False. Note that this has no effect if *RequireIssueID* is True.

Customize Issue Comments for Commits

The comments added to Gemini issues are constructed from a template file *revisionLogTemplate.txt*. The file contains an HTML table that is drawn on the browser where the Issue Comment appears. The items specific to the commit are merged with the template to form the final HTML that is sent to Gemini for the comment.

The template contains substitution tokens for the following items:

Token	Meaning
$AUTHOR	The SVN author's name
$CHECKINCOMMENT	The commit message
$FILES	The files that were affected (and how)
$REVISION	The SVN revision number assigned to the commit
$REPOSITORY	The full path or URL to the repository

Working with Sources and Debugging (Gemini-TSVN developers only!)

The integration executable is developed in Microsoft Visual Studio 2008 and the C# language. It is a console application, targeted for either 32-bit or 64-bit processors (AnyCPU) and the .NET Framework 3.5. Building is straightforward.

Debugging the integration program is easy, and you can do it right in the VS2008 IDE. To get set up:

1. Copy the files from the Hook Files folder (the two .cmd files and the revision log template) into the project's ...\bin\debug folder.
2. Edit the two .cmd files and change the path to the *CounterSoft.Gemini.SourceControl.TSVN.exe* file to point to the executable in the ...\bin\debug folder. This will cause the debug executable to be started instead of the normal one in Tortoise's Hooks folder.
3. For the mode you want to debug (pre-commit or post-commit) edit the relevant .cmd file and add "-debug" to the command. For example, in *post-comimit.cmd*, change "post-commit" to "post-commit-debug".
4. Change the Hook Scripts settings in TortoiseSVN to point to the *pre-commit.cmd* and *post-commit.cmd* files in your ..\bin\debug folder.
5. Check that the the "Wait for the script to finish" checkbox is set in the TortoiseSVN Hook Script setting.
6. Open VS2008 and load the TSVN-Gemini solution. Open the *Program.cs* file into the editor.
7. Have a test project set up as described in Set Up a Project above.

Now, when you do a commit from your test project, you'll see a "user break" Dr. Watson popup with a Debug button on it. Click Debug, and you will then be asked which debugger to use. Usually, the first one listed will be your running instance of VS2008. Select your running instance of VS2008 and click OK. Immediately you will see a programmed breakpoint get hit in the IDE and execution will stop. You're in control! You can resume from the programmed breakpoint, step past it via F10, or whatever.

You also might find the DebugView tool (described in the Troubleshooting section above) to be useful, though I find it much more efficient to work directly in the IDE compared to reading trace statements and trying to figure out where I made my mistake(s).

COPYRIGHT NOTICE