

SVN in use

Roger Burkhardt
`roger.burkhardt@fhnw.ch`

Fachhochschule Nordwestschweiz
Hochschule für Technik
Institut für Geistes- und Naturwissenschaft

2012/13



Was ist SVN

Subversion (kurz SVN) ist ein modernes **Versionsverwaltungssystem**, welches Dokumente zentral auf einem Server speichert und diese Gruppen von Nutzern zum Bearbeiten weltweit zur Verfügung stellt. Das Arbeiten mit einem solchen Versionsverwaltungssystem ist Heute beim Entwickeln grösserer Softwareprojekte die Regel. Dieses Dokument beschreibt die Verwendung von Subversion mit dem Subversion-Client **TortoiseSVN**. Dieses Dokument beschreibt nicht das Aufsetzen eines Subversion-Servers. Hierzu sei auf das folgende, sehr gute Buch verwiesen:

Literatur

- Version Control with Subversion. B. Collins-Sussman, B. Fitzpatrick, M. Pilato. <http://svnbook.red-bean.com>.

Arbeiten mit SVN

Unter Windows arbeitet man mit SVN häufig mit **TortoiseSVN**, dem klar besten Subversion-Client. TortoiseSVN (aktuell Version 1.7.9) ist OpenSource Software und kann von folgenden Quellen bezogen werden:

TortoiseSVN

<http://tortoisesvn.net/>
<http://tortoisesvn.tigris.org/>

Im Rahmen der Projekte im Studiengang Informatik an der FHNW steht ein Subversion-Server zur Verfügung. Die Zugänge sind die folgenden:

Subversion-Server der FHNW - Studiengang I

<https://projecthost.cs.technik.fhnw.ch/.../svn/>

Information für Studierende

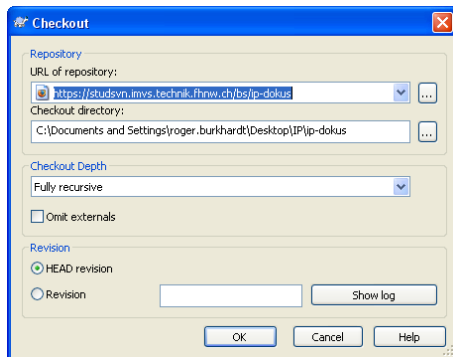
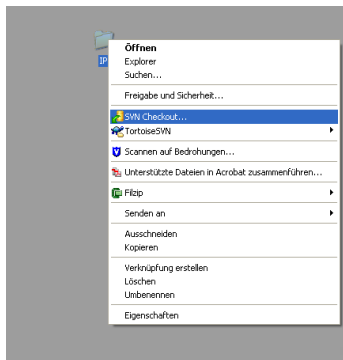
Wenn sie für ein Projekt ein Subversion-Repository benötigen, so teilen Sie dies Ihrem Dozenten oder Betreuer mit. Jedes Projekt beinhaltet automatisch ein Subversion-Repository und eine Trac-Instanz, sie müssen diese also nicht separat beantragen. Wenn Sie Ihr Repository beantragen, so sind folgende Informationen wichtig:

- Name des Projekts sowie ein Kürzel, welches später in der URL erscheint (z.B. myproject)
- Mailadressen aller Dozenten bzw. Betreuer, die Zugang haben müssen
- FHNW-Mailadressen aller Studierenden, die Zugang haben müssen
- Falls Sie Trac verwenden wollen, die Namen der Studierenden, welche Administrationszugriff auf dem Trac haben sollen

Projektarchiv (Repository) auschecken

Subversion speichert alle unter der Versionsverwaltung stehenden Dateien und Verzeichnisse im Projektarchiv (**Repository**). Die Archive für die Informatik-Projekte wurden angelegt und die entsprechenden Zugangsdaten sollten sie erhalten haben. Um nun mit dem Projektarchiv zu arbeiten, muss eine Arbeitskopie vom Server heruntergeladen werden. Legen sie dazu einen entsprechenden Ordner an, in den sie die Arbeitskopie speichern wollen. In diesem Ordner (im Windows-Explorer) kann nun mit der rechten Maustaste ein Kontextmenu geöffnet werden. In diesem Kontextmenu findet man den Befehl **SVN Checkout....** Nach dem Anwählen dieser Aktion erscheint ein Fenster, in dem die URL des Projektarchivs angegeben wird:

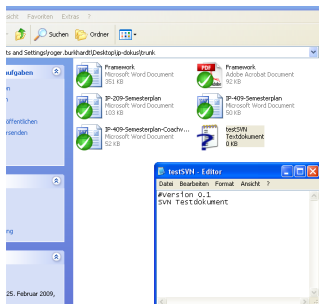
Projektarchiv (Repository) auschecken



Im angelegten Ordner wird ein neuer Ordner erstellt, in dem alle Dokumente (und Verzeichnisse) des Projektarchivs beinhaltet sind.

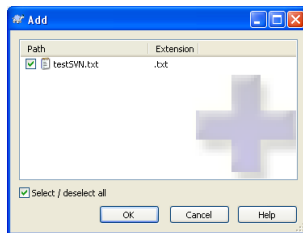
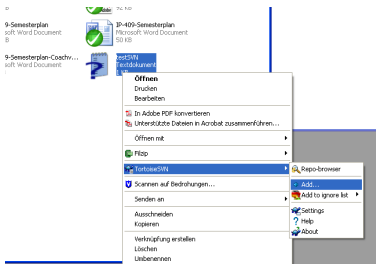
Objekte ins Projektarchiv hinzufügen

Am Anfang ist das Projektarchiv leer und es müssen Dateien und Verzeichnisse erstellt und hinzugefügt werden. Nachdem in der ausgecheckten Arbeitskopie ein Verzeichnis oder eine Datei erstellt wurde, kann dieses Objekt in zwei Schritten ins Projektarchiv geladen werden.



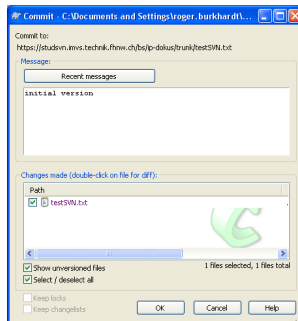
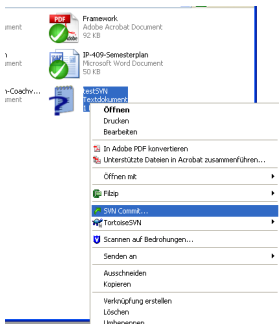
Objekte ins Projektarchiv hinzufügen

Zuerst teilt man dem Projektarchiv mit dem **Add ...**-Befehl mit, dass neue Objekte zum Projektarchiv gehören. Dazu drückt man wieder die rechte Maustaste auf dem Projektordner und wählt im Kontextmenu den Punkt **Add ...**:



Objekte ins Projektarchiv hinzufügen

Nach dem Hinzufügen der Objekte mit Add müssen die Änderungen nun noch auf den Server übertragen werden. Dies geschieht mit dem Befehl **Commit** ...:

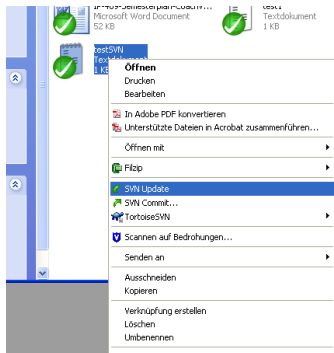


Bemerkungen

- Durch das Einchecken (Commit) erhalten die Objekte eine (neue) Revisionsnummer. Dabei erhalten alle Dokumente die gleiche Revisionsnummer. Diese Revisionsnummer bezeichnet den Stand des ganzen Projektes. Die einzelnen Objekte erhalten unabhängig von Subversion eine unabhängige Versionsnummer!
- Beim einchecken sollte immer ein kurzer Text für das Logfile eingetragen werden. Dieses Logfile ermöglicht die Suche nach bestimmten Revisionen und dokumentiert auch gleich den Verlauf des Projekts.
- Programmcode soll immer nur in lauffähigen Versionen eingcheckedt werden (wer möchte schon einen Quellcode mit Fehlern herunterladen und zuerst Fehler ausmerzen müssen!!!).

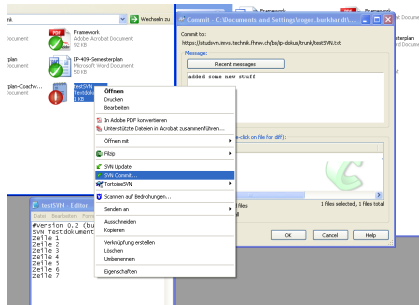
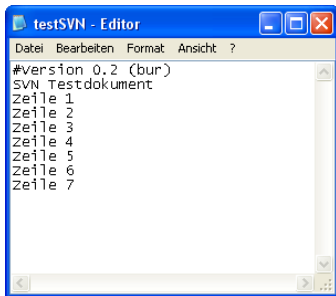
Update und Commit

Der normale Arbeitsablauf an bestehenden Dateien besteht aus dem folgenden Zyklus: **Update** → Bearbeitung der Datei → **Commit**. Dazu wird vor dem Arbeiten die neueste Revision heruntergeladen:



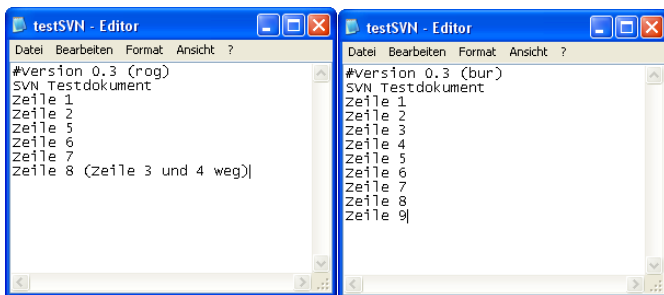
Update und Commit

Die bearbeitete Datei wird dann mit **Commit** wieder eingeecheckt:



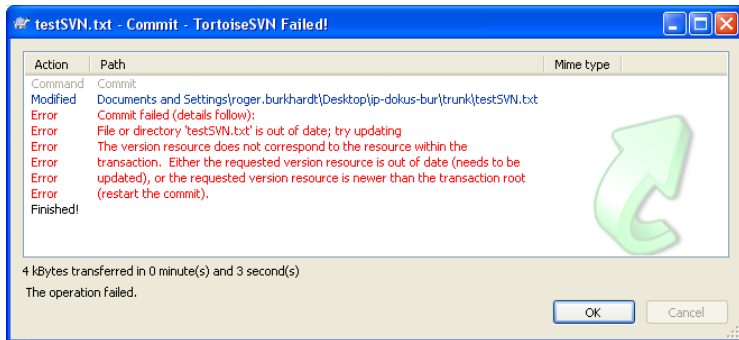
Konflikte

Der gerade beschriebene Zyklus funktioniert nicht, wenn mehrere Nutzer gleichzeitig eine Datei bearbeiten. Denken wir uns die beiden Nutzer bur und rog, welche die gleiche Datei (in der gleichen Revision) ausgecheckt haben. Seien die folgenden Änderungen an der Datei vorgenommen worden:



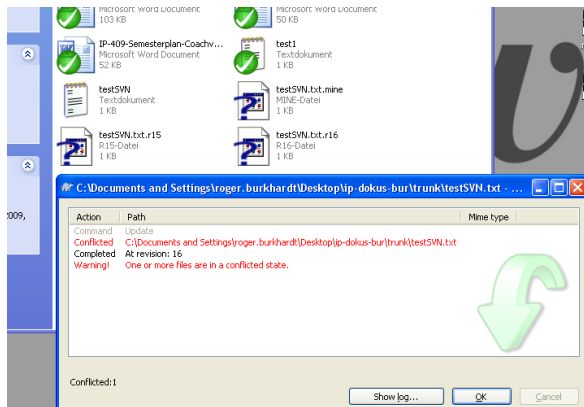
Update und Commit

Nehmen wir nun an, das der Nutzer roger die bearbeitete Datei zuerst eincheckt (commit). Dabei entstehen keine Probleme. Zu einem späteren Zeitpunkt versucht nun der Nutzer bur seine geänderte Datei hochzuladen, er erhält dabei einen Fehler:



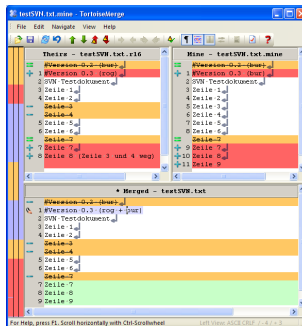
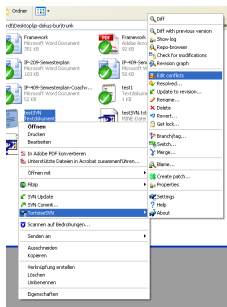
Konflikte auflösen

Um den Konflikt aufzulösen muss der Nutzer bur nun zuerst ein update durchführen. Dazu werden im entsprechenden Ordner die verschiedenen zueinander in Konflikt stehenden Dateien abgelegt:



Konflikte auflösen

Mit der Auswahl des Menüpunkts **TortoiseSVN** → **Edit Conflicts** wird ein neues Fenster mit den beiden Dateivarianten geöffnet. Hier können die beiden Dateien zusammengeführt werden:



Danach kann der Konflikt als gelöst gekennzeichnet werden und die bearbeitete Datei wieder eing检eicht werden!

Weitere Funktionen

- Löschen und Umbenennen von Objekten: Im Kontextmenu stehen die Punkte **Delete** und **Rename** für das Löschen und Umbenennen von Dateien zur Verfügung. ACHTUNG: Nie Objekte in einer Arbeitskopie direkt Löschen oder Umbenennen. Dies kann zu schweren, nicht auflösbaren Konflikten führen!
- Revisionsgraph und Revisionslog: Zu jedem Objekt kann eine grafische Darstellung der Revisionen der Datei erzeugt werden (**Revisiongraph**). In dieser grafischen Darstellung findet man auch alle Log-Einträge. Mit dem Befehl **Blame** kann zu einer Textdatei die ganze Geschichte der Entwicklung angezeigt werden (wer hat welche Änderungen vorgenommen!).

Weitere Funktionen

- Branches/Tags: Wenn von einem Softwareprojekt eine Version ausgeliefert werden soll, so sollte der Projektstand eingefroren werden. Mit dem Befehl **Branch/Tag** kann eine neue Arbeitskopie erstellt werden. Dabei wird auf dem Server eine Kopie angelegt, an welcher weiter gearbeitet werden kann.
- Merge: Werden an verschiedenen Kopien gearbeitet (entwickelt), hat man mit dem Befehl **Merge** die Möglichkeit zwei Zweige wieder zusammenzuführen (verlangt noch einiges an Handarbeit!).

Trac

In Zusammenhang mit dem von uns verwendeten Subversion-Server steht mit dem Programm **Trac** auch ein Ticketsystem zur Verfügung. Zugang auf das Programm hat man über einen beliebigen Browser:

Trac-Zugang

<https://projecthost.cs.technik.fhnw.ch/.../trac/>

Mit dem Trac-Programm kann die Arbeit in kleine Arbeitspakete verpackt werden. Diese Arbeitspakete können einem Nutzer zugeordnet werden (Ticket). Im weiteren können diese Arbeitspakete in einen Zeitplan eingebunden werden (Meilensteine). Abgearbeitete Pakete werden quittiert. Bei allen Aktionen im Trac werden die beteiligten Nutzer per Mail benachrichtigt.

Trac

Über den Browser bietet Trac die Möglichkeit die Inhalte des Projektverzeichnis zu betrachten bzw. zu durchsuchen:

Trac interface showing the file browser for the project 'ip-dokus'.

Navigation tabs: Wiki, Timeline, Roadmap, **Browse Source**, View Tickets, New Ticket, Search, Admin.

File list:

Name	Size	Rev	Age	Last Change
trunk		19	7 hours	roger.burkhardt@fhnw.ch: --
version_1.0		18	7 hours	roger.burkhardt@fhnw.ch: --
Framework.doc	351.0 kB	1	10 hours	roger.burkhardt@fhnw.ch: new files
Framework.pdf	91.9 kB	1	10 hours	roger.burkhardt@fhnw.ch: new files
IP-209-Semesterplan.doc	103.0 kB	1	10 hours	roger.burkhardt@fhnw.ch: new files
IP-409-Semesterplan-Coachversion.doc	52.0 kB	1	10 hours	roger.burkhardt@fhnw.ch: new files