

Dokumentenmanagement mit \LaTeX und Subversion

Uwe Ziegenhagen

Die Nutzung eines Versionsverwaltungssystems bietet eine Reihe von Vorteilen in der täglichen Arbeit von Programmierern und Autoren. Die Zusammenarbeit in einem Team vereinfacht sich drastisch, da der mühsame und fehlerträchtige Austausch von Dateien über FTP oder E-Mail entfällt, ältere Versionen einer Datei können problemlos wiederhergestellt werden und die Zusammenführung verschiedener Versionen vereinfacht sich. Ein nicht zu unterschätzender Vorteil ist zudem die Möglichkeit, Backups quasi im »Vorbeigehen« zu erstellen.

Mit Subversion existiert eine moderne Versionsverwaltung, die auf allen gängigen Plattformen genutzt werden kann und die wenig Konfigurations- und Einarbeitungszeit erfordert. In diesem Artikel geht es um die Nutzung von Subversion mit \LaTeX . Erläutert wird die Installation und Konfiguration auf Windows und Linux-Systemen, außerdem werden einige Pakete besprochen, die eine bequeme Integration von Subversion-Information in \LaTeX ermöglichen.

CVS und Subversion

Im Unterschied zum weit verbreiteten *Concurrent Versions System* (CVS) bezieht sich die Versionisierung von Subversion nicht auf einzelne Dateien, sondern auf ein ganzes Verzeichnis oder einen Verzeichnisbaum. Jede Versionsnummer n bezieht sich auf den Zustand dieses Verzeichnisses, nachfolgend Repository genannt, nach der n -ten Übermittlung der Daten, dem sogenannten *Commit*. Wenn also über eine Datei in Version 4 gesprochen wird, bezieht man sich auf die Datei, so wie sie in der vierten Revision existierte.

Bei jedem Check-out von Dateien aus einem Subversion-Repository wird diejenige Version ausgeliefert, deren Version kleiner oder gleich der gewünschten Versionsnummer ist. Kopien der ausgelieferten Dateien speichert Subversion in einem speziellen Verzeichnis (`.svn`) bei jedem Check-out, Update und Commit. Obwohl dies den benötigten Festplattenplatz verdoppelt, hat es doch einen entscheidenden Vorteil: Änderungen lassen sich auch ohne Netzwerkzugang

nachverfolgen und beim Commit muss Subversion nicht die gesamte Datei, sondern nur die jeweiligen Unterschiede zum Server übermitteln. CVS im Gegensatz muss die ganze Datei übermitteln, da die Änderungen serverseitig ermittelt werden. Subversion-Commits sind atomar, d. h. Änderungen an einer Datei werden entweder komplett oder überhaupt nicht gespeichert. Eventuelle Netzwerk-Probleme können also nicht zu inkonsistenten Repositories führen.

Installation

Für die Installation von Subversion gibt es verschiedene Möglichkeiten und Vorgehensweisen. Man kann entweder Svnserve [5] als eigenständigen Dienst bzw. Daemon nutzen oder aber Subversion als Apache-2-Modul installieren, das WebDAV¹, eine Erweiterung des HTTP-Protokolls nutzt.

In diesem Artikel werde ich auf beide Installationsarten unter Windows und Ubuntu Linux eingehen. Der eigenständige Dienst lässt sich zwar einfacher installieren, bei der Nutzung von Subversion als Apache-Modul gibt es aber zwei – je nach Situation nützliche – Aspekte: Man kann auf ein Repository mit dem Web-Browser zugreifen, außerdem können die in Apache eingebauten Möglichkeiten der Nutzer-Authentifizierung genutzt werden.

Windows XP

Installation als Apache-Modul

Binär-Versionen von Apache 2 sind verfügbar von [1]; persönlich bevorzuge ich aber die Nutzung einer WAMP-Umgebung² wie beispielsweise [7]. Wir entpacken xampp.zip³ nach C:/xampp und starten den Apache-Server über die xampp-control.exe. Wenn wir nun die Webseite <http://localhost> in einem Browser öffnen, sollte die Startseite, wie in Bild 1 dargestellt, angezeigt werden.

Bei einer lokalen Installation sollte man entsprechende Sicherheitsmaßnahmen gegen den Zugriff aus dem Internet ergreifen. Entsprechende Informationen findet man unter dem Punkt *security* im xampp-Menü sowie in der Apache-Dokumentation [11].

Wir laden die Binär-Version von Subversion von [4] herunter und extrahieren alle Dateien aus dem Zip-Archiv nach C:/Programme/Subversion. Nachdem

¹ Web-based Distributed Authoring and Versioning

² Windows-Apache-MySQL-PHP

³ Aktuelle Version: 1.6.7

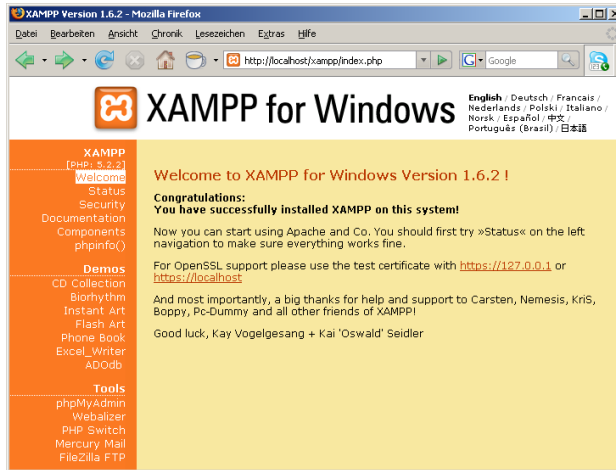


Abbildung 1: Screenshot der xampp Startseite

wir den Pfad zum Verzeichnis `C:/Programme/Subversion/bin` aus der Umgebungsvariablen `PATH` von Windows hinzugefügt haben, können wir `svn help` aus der Kommandozeile starten, um die Installation zu prüfen. Im nächsten Schritt kopieren wir `mod_authz_svn.so` und `mod_dav_svn.so` aus dem Verzeichnis `subversion/bin` in das Apache-Verzeichnis `modules` und überschreiben eventuell vorhandene ältere Versionen.

Im letzten Schritt schalten wir im Apache die WebDAV-Erweiterung und das Subversion-Modul ein, indem wir folgende Zeilen der `httpd.conf` im Apache-Verzeichnis `/conf` hinzufügen:

- `LoadModule dav_svn_module modules/mod_dav_svn.so` und
- `LoadModule authz_svn_module modules/mod_authz_svn.so`

Bevor wir Apache neu starten, ist der allerletzte Schritt die Erstellung des Stamm-Verzeichnisses für alle Repositories `c:/allMyRepositories` und die Ergänzung der `httpd.conf` um Listing 1:

Listing 1: Setup code for the Windows repository root

```
<Location /svn>
DAV svn

SVNParentPath c:/allMyRepositories
</Location>
```

Mit der Kommandozeile wechseln wir nach `c:/allMyRepositories` und erstellen unser erstes Repository mittels `svnadmin create test`.

Wenn wir jetzt `http://localhost/svn/firstsample/` im Browser öffnen, sollten wir ein leeres Verzeichnis-Listing mit der Überschrift **Revision 0:** / sehen. Die grundlegende Subversion-Installation ist damit abgeschlossen, durch die Installation eines grafischen Subversion-Clients wie TortoiseSVN lässt sich aber die Arbeit mit Subversion deutlich erleichtern.

TortoiseSVN [6] ist ein freier Subversion-Client für Windows, der seine Funktionen über das Menü der rechten Maustaste zugänglich macht.

Das TortoiseSVN-Interface fügt sich komplett in den Windows Explorer ein; Icons zeigen in einem Arbeitsverzeichnis sofort an, welche Dateien und Verzeichnisse verändert wurden und in das Repository übermittelt werden müssen. Die Installation ist einfach, nach dem Neustart des Systems finden wir im Kontextmenü die entsprechenden Menüeinträge. Neben TortoiseSVN gibt es noch eine Reihe weiterer Clients, zum Beispiel RapidSVN (Windows, Unix/Linux) und SVNcommander (Linux), Entwicklungsumgebungen wie Netbeans und Eclipse bieten auch eingebauten Subversion-Support.

Installation als Dienst

Die Installation von `svnserve` unter Windows ist in wenigen Schritten erledigt. Wir laden den Subversion-Installer von `http://subversion.tigris.org` herunter und führen die Installation durch. Anschließend legen wir ein **Repository**-Verzeichnis an (hier `c:/repos`) und erstellen über die Kommandozeile – oder gegebenenfalls TortoiseSVN – ein erstes Repository: `svnadmin create test` und ändern die Lese- und Schreibrechte in der Datei `conf/svnserve.conf` gemäß Listing 5, um eventuelle Fehler leichter eingrenzen zu können. Über die Kommandozeile starten wir dann auch den `svnserve`-Dienst mittels `svnserve.exe -d -r c:/repos`. Per TortoiseSVN oder Kommandozeile (`svn co svn://localhost/test` sollte ein Check-out jetzt möglich sein).

Weitere eventuell notwendige Pakete werden durch das Paketmanagement selbstständig ausgewählt und konfiguriert. Nach der Installation sind die letzten Schritte die Erstellung eines Stammverzeichnisses für unsere Repositories, hier beispielsweise in `/home/uwe/repositoryRoot`, die entsprechende Anpassung von `/etc/apache2/sites-available/default` und das Setzen der notwendigen Rechte mittels `chmod -R 777 /home/uwe/repositoryRoot`:

Listing 3: Anpassung für die Datei `/etc/apache2/sites-available/default`

```
<Location /svn>
DAV svn

SVNParentPath /home/uwe/repositoryRoot
</Location>
```

In `/home/uwe/repositoryRoot` erstellen wir dann per `svnadmin create test` ein erstes Repository, das unter der URL `http://localhost/svn/test` im Browser angezeigt werden sollte. Gibt es bei diesem Schritt einen Zugriffsfehler, sind vermutlich die Rechte nicht richtig gesetzt worden.

Erste Schritte

Um unser Repository zu füllen, erstellen wir in einem Verzeichnis (alle enthaltenen Dateien werden gleich importiert) ein kleines \LaTeX -Dokument (`article-template.tex`):

Listing 4: Eine einfache \LaTeX -Datei

```
\documentclass{article}
\begin{document}
  Hello World!
\end{document}
```

Mittels Kommandozeile (`svn import http://localhost/svn/firstsample/ -m 'Anfangsimport'`) oder dem entsprechenden Eintrag im TortoiseSVN-Kontextmenü können wir jetzt die Datei importieren, 'Anfangsimport' ist der optionale Parameter für Kommentar beim Import.

Apache zeigt jetzt **Revision 1:** / im Browser (siehe Bild 3). Um jetzt an dieser Datei Änderungen vornehmen zu können, müssen wir sie in ein **Arbeitsverzeichnis** auschecken. Alle weiteren Commits werden aus diesem Verzeichnis ausgeführt.

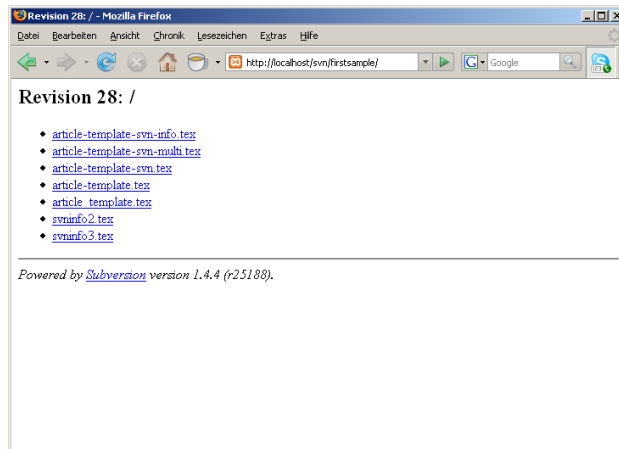


Abbildung 3: Repository mit Apache und dem Subversion-Modul

Installation als Dienst

Die Installation von `svnserve` ist noch deutlich einfacher als die Installation des Apache-Moduls. Mittels `sudo apt-get install subversion` wird die Subversion-Installation gestartet, Ubuntu lädt und installiert dann alle benötigten Pakete. Aus Sicherheitsgründen ist es ratsam, den `svnserve`-Dämon nicht unter dem `root`-Account laufen zu lassen. Wir werden daher einen Nutzer `svn` anlegen, der in seinem Home-Verzeichnis die Repository-Verzeichnisse enthält:

- `sudo useradd svn` legt den Nutzer `svn` an,
- `sudo passwd svn` setzt Passwort für den Nutzer,
- `sudo mkdir /home/svn` erstellt Home-Verzeichnis,
- `sudo chown -R svn /home/svn` setzt die korrekten Besitzrechte.

Nachdem die entsprechenden Verzeichnisse und Rechte gesetzt sind, kann jetzt das erste Repository angelegt werden. Wir wechseln per `su - svn` zum Nutzer `svn` und erstellen in dessen Homeverzeichnis das Stammverzeichnis, das die Repositories enthalten wird: `mkdir repos`. In diesem Verzeichnis nut-

zen wir `svnadmin create test`, um die von Subversion genutzten Datei- und Ordnerstrukturen anzulegen.

Für das erste Beispiel ist es sinnvoll, die Rechte bewusst so zu setzen, dass jedermann Lese- und Schreibrechte auf dieses Repository hat. Eventuelle Probleme lassen sich so leichter erkennen. Dazu bearbeiten wir die Datei `svnserve.conf`, die in `test/conf` liegt, gemäß Listing 5. In der späteren Arbeit mit Subversion, insbesondere wenn mehrere Autoren an einem Projekt beteiligt sind, ist das Anlegen von Nutzerkonten ratsam.

Listing 5: geänderter Abschnitt aus `svnserve.conf`

```
[general]
### These options control access to the repository for unauthenticated
### and authenticated users. Valid values are "write", "read",
### and "none". The sample settings below are the defaults.
anon-access = write
auth-access = write
```

Im letzten Schritt starten wir den Dienst mit `svnserve -d -r /home/svn/repos` und importieren eine beliebige Datei in unser Repository mittels `svn import -m 'initial import' svn://localhost/test`, die wir dann in ein beliebiges Verzeichnis auschecken können: `svn co svn://localhost/test`.

Ebenso wie unter Windows ist es ratsam, den Subversion-Dienst automatisch starten zu lassen oder die Kommunikation über SSH zu verschlüsseln, siehe dazu die Hinweise unter [5].

Integration mit \LaTeX

Um Subversion in unseren \LaTeX -Workflow zu integrieren, müssen wir in die \TeX -Datei entsprechende Meta-Informationen einfügen. Die folgende Liste enthält die verfügbaren Schlüsselwörter und ihre Beschreibung:

Date (*LastChangedAt*) Datum und Zeit des letzten Checkin,

Revision: (*LastChangedRevision*) Revisionsnummer,

Author: (*LastChangedBy*) Name des Autors,

HeadURL: URL der Datei,

Id: Zusammenfassung der anderen Schlüsselwörter.

Wenn man mit der Kommandozeile in das Arbeitsverzeichnis wechselt und diese Schlüsselwörter mittels `svn propset svn:keywords 'Date HeadURL Revision Id' article_template.tex` zur Expansion freigibt, erweitert Subversion diese in der Datei `article_template.tex`, jeweils eingeschlossen in `$`-Zeichen. In TortoiseSVN lassen sich die Schlüsselwörter auch bequem über den Menü-Punkt **Properties** einfügen.

Listing 6: Die Beispieldatei mit expandierten Schlüsselwörtern

```
% $Revision: 10 $
% $HeadURL: http://tools.assembla.com/svn/svnArticle/svnArticle.tex $
% $Date: 2007-07-22 19:50:04 +0200 (So, 22 Jul 2007) $
% $Author$
% $Id: svnArticle.tex 10 2007-07-22 17:50:04Z uweziegenghagen $

\documentclass{article}
\begin{document}
  Hello World!
\end{document}
```

Alle L^AT_EX-Pakete, die ich im weiteren Artikel vorstellen werde, basieren auf der Auswertung dieser Schlüsselwörter.

svn

Das Paket `svn` gestattet den Zugriff auf die Subversion-Informationen über die Syntax `\SVN $Keyword: <metadata>$`. Wenn die Schlüsselwörter korrekt expandiert wurden, stehen die folgende L^AT_EX-Befehle zur Verfügung:

- `\SVNDate` Datum des letzten Commit, `\SVNTime` als Commit-Zeit und `\SVNRawDate` als Datum und Zeit im Rohformat, wenn `Keyword $Date$` war;
- `\SVNKeyword` sonst (Beispiele: `\SVNId`, `\SVNHeadURL`)

Listing 7: Beispieldatei mit eingefügten `svn`-Befehlen

```
\documentclass{article}
\usepackage{svn}

\SVN $Id: svnArticle.tex 10 2007-07-22 17:50:04Z uweziegenghagen $
```

```

\SVN $Date: 2007-07-22 19:50:04 +0200 (So, 22 Jul 2007) $
\SVN $Id: svnArticle.tex 10 2007-07-22 17:50:04Z uweziegehenhagen $
\SVN $HeadURL: http://tools.assembla.com/svn/svnArticle/svnArticle.tex $

\begin{document}

\SVNDate \\
\SVNRawDate \\
\SVNTime \\
\SVNId \\
\SVNHeadURL
\end{document}

```

```

      July 15, 2007
2007-07-15 17:33:30 +0200 (So, 15 Jul 2007)
17:33:30
article-template.tex 12 2007-07-15 15:33:30Z
http://localhost/svn/firstSample/article-template.tex

```

Abbildung 4: Ausgabe von article-template.tex mit dem Paket `svn`

svninfo

Das Paket `svninfo` bezieht die Versionsinformationen ausschließlich aus dem Tag `Id`, das bei der Expansion von `\svnInfo` erzeugt wird: `\svnInfo $Id:`
`article-template-svn-info.tex 18 2007-07-15 16:11:21Z$`

Um die SVN-Informationen nutzen zu können, werden die folgenden Kommandos definiert:

- `\svnInfoFile` Dateiname,
- `\svnInfoRevision` Revisionsnummer,
- `\svnInfoDate` Datum des letzten check-in,
- `\svnInfoTime` Zeit des letzten check-in,
- `\svnInfoYear` Jahr wie in `\svnInfoDate`,

- `\svnInfoMonth` Monat wie in `\svnInfoDate`,
- `\svnInfoDay` Tag wie `\svnInfoDate`,
- `\svnInfoOwner` Besitzer der Datei,
- `\svnToday` Datum des letzten Check-in im `\today` Format,
- `\svnInfoMinRevision` minimale Revision des gesamten Dokuments,
- `\svnInfoMaxRevision` maximale Revision des gesamten Dokuments.

`\svnInfoMinRevision` und `\svnInfoMaxRevision` sind nützlich für Dokumente, die aus mehreren Dateien bestehen. Das Paket bietet noch weitere Optionen wie `fancyhdr`, `eso-foot`, `scrpage2`, um Subversion-Informationen am Rand oder in der Fußzeile des Dokuments auszugeben. Details dazu finden sich in der Dokumentation des Pakets.

svn-multi

Das Paket `svnmulti` stellt zwei Kommandos bereit, `\svnid` und `\svnidlong`, um Subversion-Informationen zu verwerten. Zur Ausgabe im Dokument werden folgende Makros definiert:

- `\svnrev` Revisionsnummer,
- `\svndate` Datum des letzten Check-in,
- `\svnauthor` Autor,
- `\svnfilerev` Revision der aktuellen Datei, falls dieses einen `\svnid` oder `\svnidlong` Befehl enthält oder die entsprechenden Werte der letzten Datei, falls keiner dieser Befehle in der aktuellen Datei enthalten ist.
- `\svnmainurl` und `\svnmainfilename` geben die URL beziehungsweise den Namen der Hauptdatei aus, wenn diese am Ende der Dokumentenpräambel mit `\svnmainfile` festgelegt wurde.

Weiterhin nutzt das Paket `\svn{Schlüssel}` und `\svnk{Schlüssel}`, um Subversion-Schlüsselwörter direkt auszugeben. Um auf Datumsinformationen zuzugreifen, stellt das Paket noch eine Reihe weiterer Makros bereit (die Bedeutung erschließt sich aus dem Namen): `\svnfileyear`, `\svnfilemonth`, `\svnfileday`, `\svnfilehour`, `\svnfileminute`, `\svnfilesecond`, `\svnfiletimezone`, `\svnyear`, `\svnmonth`, `\svnday`, `\svnhour`, `\svnminute`, `\svnsecond` und `\svntimezone`.

Zusammenfassung

Dieser Artikel beschreibt die Installation von Subversion unter Linux und Windows und stellt drei Pakete vor, um Subversion-Informationen in L^AT_EX-

Dokumente zu integrieren. Mehr Informationen findet sich in den Dokumentationen der einzelnen Pakete sowie in der Literatur zu Subversion ([10, 13]. Ein Feedback zu diesem Artikel, ist jederzeit willkommen, Aktualisierungen und Errata werde ich unter <http://www.uweziegenghagen.de/latex/> verfügbar machen.

Literatur

- [1] *Apache 2 web server*; <http://httpd.apache.org>.
- [2] *psvn*; http://www.xsteve.at/prg/vc_svn/.
- [3] *RapidSVN*; <http://rapidsvn.tigris.org>.
- [4] *Subversion (Software)*; <http://subversion.tigris.org/>.
- [5] *Svnserve Based Server*; http://tortoisesvn.net/docs/nightly/TortoiseSVN_en/tsvn-serversetup-svnserve.html.
- [6] *TortoiseSVN*; <http://tortoisesvn.tigris.org>.
- [7] [apachefriends.org](http://www.apachefriends.org/): *Xampp*; <http://www.apachefriends.org/>.
- [8] Paul A. Blaga: *PracTeX Journal: Making an Electronic Journal with web tools, Wiki, and version control*; *The PracTeX Journal*; 2; 2007; <http://tug.org/pracjourn/2007-2/blaga/>.
- [9] Achim D. Brucker: *svninfo*; <http://www.ctan.org/tex-archive/macros/latex/contrib/svninfo/>.
- [10] Ben Collins-Sussman, Brian W. Fitzpatrick und C. M. Pilato: *Version Control with Subversion. Next Generation Open Source Version Control*; O'Reilly; 2004.
- [11] Apache Foundation: *Apache HTTP Server Version 2.2 Documentation*; <http://httpd.apache.org/docs/2.0/en/>.
- [12] Richard Lewis: *svn*; <http://www.ctan.org/tex-archive/macros/latex/contrib/svn/>.
- [13] Mike Mason: *Pragmatic Version Control Using Subversion*; Pragmatic Programmers LLC.; 2006.
- [14] Martin Scharrer: *svn-multi*; `vormals svnkw`.
- [15] wikipedia.org: *Subversion_(software)*; in German.