



# Migration von CVS zu Subversion mit SVNpserver

Dr. Ralf Schlatterbeck  
Open Source Consulting

Email: [office@runtux.com](mailto:office@runtux.com)  
Web: <http://www.runtux.com>  
Tel. +43/650/621 40 17



## Inhalt

CVS Probleme – warum migrieren? . . . . .	3
Versionskontrolle: Kandidaten . . . . .	4
Subversion . . . . .	5
Migration von CVS zu Subversion . . . . .	6
ACHTUNG: Problembereiche . . . . .	7
Vorgehen bei der Konversion . . . . .	8
Legacy: CVS-Abhängigkeiten . . . . .	9
Solution: SVNpserver . . . . .	10
Anforderungen an SVNpserver . . . . .	11
Projekt-Status . . . . .	12



## CVS Probleme – warum migrieren?

CVS unterstützt einige essentielle Features nicht:

- Umbenennen von Dateien
- Umbenennen von Verzeichnissen
- Löschen von Verzeichnissen
- Changesets
- Atomic Commits



## Versionskontrolle: Kandidaten

Centralized ...

- CVS
- Subversion

... vs. Distributed

- Arch
- GIT
- Mercurial (HG)



## Subversion

- Einfache Verwendung wenn von CVS migriert wird
- Zentrale Versionskontrolle funktioniert gut in zentralem Entwicklerteam
- Entscheidung (vgl. [SVN-FAQ](#))
  - Ein Repository mit mehreren Projekten
  - getrenntes Repository für jedes Projekt

Wenn Migration von Dateien zwischen Projekten zu erwarten ist (Refactoring, Auslagern in Lib)  
→ single Repository



## Migration von CVS zu Subversion

Aus der Dokumentation von [cvs2svn](#):

- Top-Skimming – kein cvs2svn ab einem Stich-Zeitpunkt SVN
- Trunk only – keine Tags, keine Branches
- Pick & choose – was wenn etwas vergessen wird?
- Full Conversion
- One project at a time



## ACHTUNG: Problembereiche

- Zeilenende Markierungen: In SVN automatisch
- Binäre Dateien: CVS unterscheidet nicht!
  - CVS *kann* Dateien als binär markieren
  - SVN *braucht* diese Unterscheidung
  - Repositories unter Unix/Linux mit Unix/Linux Clients haben sicher nicht alle Binärdateien als binär markiert!

*Falls versehentlich eine Binärdatei als Text importiert wird, ist der Inhalt mit allen Revisionen in SVN kaputt!!*



## Vorgehen bei der Konversion

- Kommandozeile vs. Konfigurationsdatei
- Inkonsistente Branches/Tags in CVS
- Verifikation: Vergleich Checkout von CVS und SVN  
Nach Möglichkeit einzelne (getagte) Versionen
- Konfigurationsdatei bzw. cvs2svn Aufruf unter Versionskontrolle bis alles passt



## Legacy: CVS-Abhängigkeiten

Was tun mit CVS-Abhängigkeiten in bestehenden Anwendungen?

- Build-Systeme die auf CVS zugreifen
  - Portierung nicht wirtschaftlich
  - Besonders bei selten benötigten Branches
- Software-Werkzeuge die mehrere closed-source Versionskontrollsysteme unterstützen, aber im Open-Source Bereich nur CVS verwenden können
  - Verzicht/Umstieg nicht möglich
- ...



## Solution: SVNpserver

- Zugriff auf Subversion Repository mit CVS Mitteln
- Spricht CVS pserver-Protokoll
- Backend ist Subversion Repository
- Standard-CVS Kommandos funktionieren mit Subversion
- andere Tools (nicht CVS) können CVS-Schnittstelle verwenden



## Anforderungen an SVNpserver

- Unterstützung der wichtigsten CVS Kommandos add, remove, commit, diff, tag
- Verschiedene SVN Repository Strukturen (Single-Project, Multi-Project)
- einfache Installation

Nicht-Ziele

- Performance
- Verwendung von CVS-Kommandos statt SVN



## Projekt-Status

- Vier Releases seit Dez 2007
- Tested & Works with CVS commands
- Derzeit: Testen mit einem Design-Werkzeug das nur CVS unterstützt
- Wird in Kürze für eine große CVS zu Subversion Migration in Produktion gehen
- wohl noch einige Bug-Fix Releases: CVS-Protokoll ist sehr schwammig definiert

Verfügbarkeit: [svnserver.sourceforge.net](http://svnserver.sourceforge.net)



## Links

- Homepage von cvs2svn, CVS to Subversion Repository Converter: [cvs2svn.tigris.org](http://cvs2svn.tigris.org)
- svnpserver: A CVS pserver implementation for Subversion [svnpserver.sourceforge.net](http://svnpserver.sourceforge.net)
- How do I manage several different projects under Subversion? [Subversion-FAQ](#)