

# CM Paper

## Neues in Subversion 1.7

**WC-NG**

**HTTPv2**

**Und mehr**

## Neues in Apache Subversion 1.7

Das Open Source Projekt Apache Subversion [1], das bekannte und beliebte Werkzeug für Versionskontrolle, geht in den Endspurt zum Release 1.7. Hier ein Blick auf die Neuerungen und Änderungen, die wir diesen Sommer erwarten können.

### Einleitung

Das Aushängeschild der Version 1.7 ist *wc-ng* (Working Copy, the Next Generation), aber auch viele weitere größere und kleinere Features haben es ins Subversion Release 1.7 geschafft. So zum Beispiel:

- `svn patch`, das Gegenstück zu `svn diff`
- HTTPv2, das optimierte HTTP-Protokoll
- Neue serverseitige Tuning-Möglichkeiten durch den neuen Repository Cache und Einstellung der Netzwerk-Kompressionsrate.

Die Subversion Client API und die `svn` Kommandozeile verhalten sich nun in vielen Randfällen sinnvoller und sicherer. Das zeigt sich u.a. durch

- Korrekturen an File Externals,
- komfortableres Mergeinfo und
- verbesserten Umgang mit verschachtelten Operationen und gelöschten oder verlorengegangenen Verzeichnisbäumen.

Die Liste der Neuerungen ist damit noch nicht erschöpft – wohl kaum eine Überraschung, denn seit Subversion 1.6.0 sind ja nun mehr als zwei Jahre vergangen. Die folgenden Darstellungen beschäftigen sich mit einigen der herausragenden Änderungen. Die vollständige Liste der neuen Features findet sich, wie immer, in den Release Notes [2].

### WC-NG

Werfen wir zunächst einen Blick auf die größte Neuentwicklung im Subversion 1.7 Release. Die Working Copy Library (*libsvn\_wc*), bisher das älteste Stück Quellcode des Subversion Projekts, verwaltet die Arbeitskopien, die Subversion für jeden Benutzer anlegt. Die alte *libsvn\_wc* bereitet Entwicklern neuer Features schon längere Zeit Kopfschmerzen. Vor allem das noch von CVS übernommene Konzept, die Metadaten in Form vieler `.svn`-Verzeichnisse in der ganzen Arbeitskopie zu verteilen, erzeugt besonders auf Windows übermäßigen I/O Overhead im Dateisystem und birgt viele Tücken für die Verwaltung von nicht mehr (oder noch nicht) vorhandenen Verzeichnisbäumen. Viele Randfälle verhielten sich deshalb bisher problematisch und teils auch fehlerhaft, zum Beispiel:

- das Ersetzen von einer Datei mit einem Verzeichnis und umgekehrt (*replace-by-different-kind*),
- bestimmte Fälle ineinander verschachtelter Umstrukturierungen von Verzeichnisbäumen (*nested moves/copies/deletes/adds*) und
- die Aufzeichnung von Tree Conflicts generell.

Das mit hohen Erwartungen verbundene Feature *wc-ng* ist aber hauptsächlich hinter der *libsvn\_wc* API verborgen. Das heißt, obwohl dessen Inneres von Grund auf umstrukturiert wurde, sieht die neue *libsvn\_wc* nach außen hin kaum anders aus. Die wohl sichtbarste Veränderung für Subversion-Benutzer ist die Verschmelzung der bisher vielen verteilten `.svn` Meta-Verzeichnisse zu einem einzigen `.svn` Verzeichnis an der Wurzel der Arbeitskopie, ganz wie es zum Beispiel Git und Mercurial schon vorgemacht haben. In diesem zentralen `.svn` Verzeichnis findet sich erstens die nagelneue `wc.db` Datei, eine SQLite Datenbank für die Metadaten der gesamten Arbeitskopie, und zweitens das `pristines` Verzeichnis, in dem die zu den Metadaten gehörigen Dateiinhalte abgelegt sind (siehe Abb. 1).

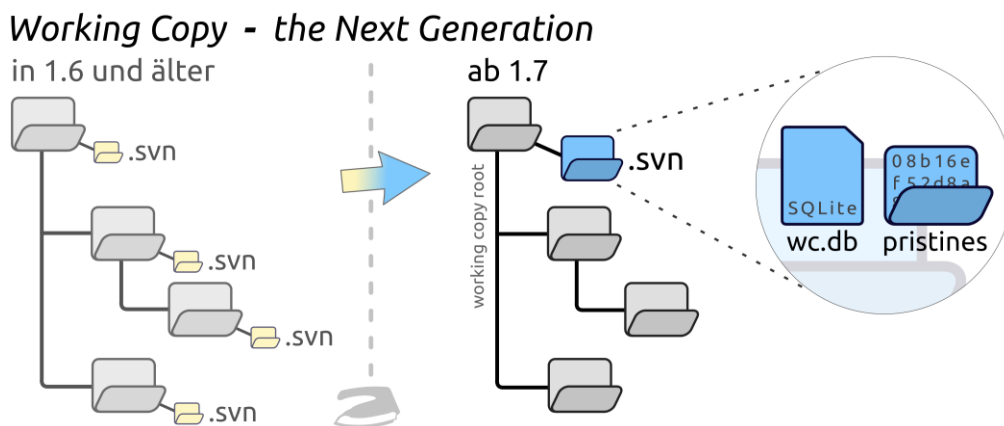


Abbildung 1. Vollständig neue Struktur der Metadaten in Arbeitskopien ab Subversion 1.7

Mit der `wc.db` Datenbank wird u.a. der Löwenanteil des bisher nötigen Datei-Locking durch Datenbank-Transaktionen abgelöst. Die neuen Datenstrukturen der SQLite Datenbank sind außerdem von Grund auf für all jene Fälle ausgelegt, die man im Jahre 2000 (zu CVS-Zeiten) einfach noch nicht wichtig fand, die jedoch inzwischen immer wieder Probleme bereiten. Versierte Anwender müssen der Verlockung nicht widerstehen, mit direkten SQL-Queries selbst mal in den Metadaten nachzuschauen; Tool-Entwickler jedoch sollten weiterhin ausschließlich die *libsvn\_wc* API verwenden, um von Subversions wohlgepflegter Rückwärts-Kompatibilität profitieren zu können, die auf der internen Datenbankebene natürlich nicht gewährleistet ist.

Der `pristines` Ordner ergänzt die SQLite Datenbank. Er ist ein durch SHA1-Checksummen indizierter Speicher für unangetastete, meist vom Repository heruntergeladene Volltext-Dateiinhalte. Das `pristines` Verzeichnis erinnert ebenfalls entfernt an die internen Strukturen von Git und Mercurial: Jeder Dateiinhalt wird in einer Datei abgelegt, deren Name die SHA1-Checksumme des Dateiinhalts selbst ist. Identische Dateien werden also inhärent sofort erkannt und nur einmal gespeichert. Anders als bei Mercurial/Git sind diese SHA1-indizierten Dateien aber völlig frei von Metadaten.

*Wc-ng* sollte nicht nur alten, gewachsenen Spaghetticode aufräumen, sondern im gleichen Zuge auch die Verwaltung von Arbeitskopien viel schneller machen. Einer der Gründe für den außerordentlich langen Release Cycle für 1.7.0 war, dass ebendiese erwartete dramatische Laufzeitverbesserung zunächst ausblieb. Einige verblüffend effektive Optimierungen der neuen zentralisierten Working Copy brachten dann aber seit etwa April dieses Jahres die erhoffte Performance im Working Copy Layer. Entsprechende Test Suites zeigen inzwischen eine

durchgängige Beschleunigung aller zeitlich ausschlaggebenden lokalen Verwaltungsvorgänge in Subversion 1.7 Clients, teils sogar fünffach schneller als Subversion 1.6 [3] – hiermit ist aber nur der lokale Overhead in den Client-Bibliotheken gemeint. Am verhältnismäßig langsamen Netzwerk gemessen wird man mit Sicherheit keine vielfache Beschleunigung feststellen können.

Von Anfang an konnte *wc-ng* in sehr tiefen Arbeitskopien überzeugen. Denn noch in Subversion 1.6 explodierten bei besonders tief verschachtelten Verzeichnissen im Client sowohl der Speicherbedarf als auch die Laufzeit, was so manchem kommerziellen Anwender Sorgen bereitete. Dieses Problem ist ab Subversion 1.7.0 vollständig behoben. In einer einfachen Testumgebung mit 100 ineinander liegenden Unterverzeichnissen war *wc-ng* auf Anhieb fast dreimal so schnell wie die alte *libsvn\_wc*. Insbesondere aber zeigt Subversion 1.6 in bestimmten Fällen eine von der Pfadtiefe abhängige Explosion bis hin zu hunderten Megabyte Arbeitsspeicherbedarf, und könnte durchaus einen Rechner per *swap death* in die Knie zwingen. Mit 1.7 bleibt in diesen Fällen der Speicherbedarf konstant bei verschwindend wenigen Megabytes.

Über die Version 1.7 hinaus sind große Ziele geplant, für die *wc-ng* bereits ausgelegt ist. Voraussichtlich wird jeder Benutzer schließlich nur noch insgesamt ein einziges `.svn` Meta-Verzeichnis im Heim-Verzeichnis haben, von dem eine beliebige Anzahl verschiedener Arbeitskopien von verschiedenen Repositories zentral verwaltet werden. So teilen sich mehrere Arbeitskopien denselben `pristines` Ordner, der bei wiederholten Checkouts derselben Daten nicht erneut vom Server heruntergeladen werden muss. Neue Arbeitskopien könnten dann also oft in Windeseile, mit kaum oder gar keinem Serverkontakt erstellt werden. Das `pristines` Verzeichnis bietet auch eine saubere Infrastruktur, um in Zukunft Features wie Offline-Commits (*shelving*) und intelligentes lokales Repository-Mirroring zu implementieren. Das und vieles mehr ist aber heute noch Zukunftsmusik und wird allerfrühestens für Subversion 1.8.0 infrage kommen.

Trotz der zahlreichen Verbesserungen werden einige Benutzer aufstöhnen, wenn sie Subversion 1.7 ausprobieren. Und zwar genau jene, die bisher gern statt einem neuen `checkout` einfach einen Unterbaum einer bereits bestehenden Arbeitskopie kopiert oder verschoben haben, um Serverkontakt zu vermeiden. Da es ab 1.7 nur noch ein zentrales `.svn` Verzeichnis per Arbeitskopie gibt, kann man nur noch das Wurzelverzeichnis der Arbeitskopie als Ganzes effektiv kopieren und verschieben. Die Unterverzeichnisse sind nun nichts weiter als Dateiinhalte bar jeder Metadaten. Der einzige in 1.7 existierende Workaround für diesen zufällig entstandenen Use Case wäre also, die Arbeitskopie als Ganzes zu kopieren. Hierbei sollte man unbedingt sichergehen, dass während der Kopie kein Subversion-Client-Prozess auf die Arbeitskopie zugreift, um Datenkorruption der `wc.db` zu vermeiden. Wem das nicht genügt, kann natürlich, wie immer, auch weiterhin Subversion 1.6 Clients nutzen, um auf Subversion 1.7 Repositories zuzugreifen.

## HTTPv2

WebDAV galt zur Jahrtausendwende als die glänzende Neuerung der webbasierten Dateiverwaltung, und Subversions HTTP-Protokoll wurde zukunftsweisend WebDAV-kompatibel entworfen. Um den WebDAV-Richtlinien gerecht zu werden nahm Subversion sogar ein wenig überflüssige Kommunikation zwischen Server und Client in Kauf (Stichwort *http-roundtrips* [4]). In der Praxis wurde WebDAV aber nur von wenigen Tools aufgegriffen und ist nun eher Hindernis als Hilfe in der Server-Client Kommunikation. Für Subversion 1.7 wurde daher das

HTTP-Protokoll überarbeitet, um u.a. ebendiese überflüssigen *roundtrips* zu eliminieren. Sobald nun Server und Client im Kontakt über HTTP feststellen, dass beide Seiten das neue, optimierte HTTPv2-Protokoll verstehen, schalten beide auf dieses um. Die Kommunikation mit älteren Servern oder Clients läuft wie gehabt.

Die Server-Client Kommunikation per HTTP -URL kann sich also beschleunigen, sobald beide Seiten mindestens Subversion 1.7 einsetzen, und zwar insbesondere bei hoher Netzwerk-Latenz.

## svn patch

Mit dem neuen Release wird auch der `svn patch` Unterbefehl eingeführt (Abb. 2). Hier schließt sich eine Lücke in Subversions Befehlskette: während Unix-Benutzer bisher das mitgelieferte `patch` Programm nutzen würden, um per `svn diff` erstellte Patches in eine Arbeitskopie einzuspielen, fehlt auf Windows dieser Standard. Deshalb mussten Windows-Werkzeuge wie TortoiseSVN bislang eigene Patch-Mechanismen zur Verfügung stellen.

Doch gerade bei Unix `patch` fehlt jede Integration mit Subversion. Wird beim Patchen eine Datei erstellt oder gelöscht, muss man dies bislang zusätzlich per `svn add` und `svn delete` manuell der Working Copy mitteilen. Das komplette Fehlen aller Property-Änderungen, Verschiedenartige Zeilenumbrüche, Subversion-Keywords und Pfadbaum-Umstrukturierungen sind weitere Unhandlichkeiten, die im Umgang mit Patch-Dateien bisher stören. Der neue `svn patch` Befehl wird leidenschaftlichen "Patchern" große Freude bereiten, da er die eben genannten Probleme weitgehend automatisch meistert und somit erstmals einen Workflow zulässt, der einem "Offline Commit" (oder *shelving*) gleicht. Die bessere Integration macht sich auch bei Tortoise & Co. bemerkbar.

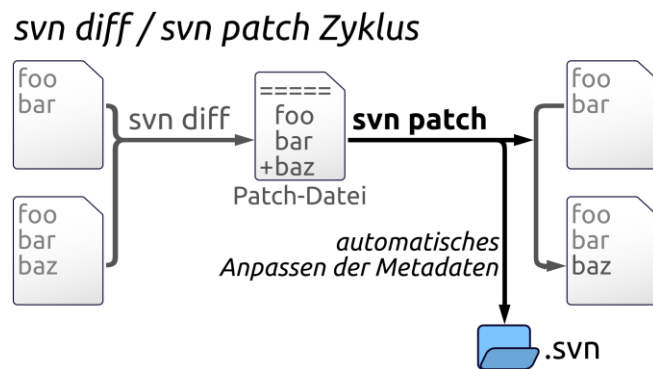


Abbildung 2. Patcher's Heaven – der neue `svn patch` Befehl

Die erste Implementierung, enthalten im Subversion Release 1.7, hat folgende Fähigkeiten:

- `svn patch` vermeidet Textkonflikte, die allein von Subversion Keywords herrühren (`$Author:$`, `$Date:$`, etc.),
- gleicht Zeilenumbruch-Standards automatisch ab und

- leitet vom herkömmlichen Patch-Inhalt entsprechende `svn add` und `svn delete` Operationen ab, die dann automatisch der Subversion Arbeitskopie mitgeteilt werden. Entleerte Verzeichnisse werden so auch automatisch entfernt.
- Selbst Änderungen an Subversion-Properties, also Änderungen per `svn propset/propedit/propdel`, werden unterstützt. (Für das Patchen von Properties muss das Diff mit Subversion 1.7 erstellt worden sein.)

In zukünftigen Subversion Releases soll das Angebot der `svn diff` → `svn patch` Kommandokette dann weiter ausgebaut werden:

- volle Unterstützung für Git-Diffs,
- explizites `copy` und `move`, und
- explizites Erstellen und Löschen von Dateien und Verzeichnissen (statt dem impliziten Löschen in 1.7).

Es entsteht also ein plattformübergreifender Standard für Subversion-Patches, der direkt in Subversion integriert ist und so von der Notwendigkeit für externe Patch-Tools befreit, und der außerdem den Umgang mit Patch-Dateien in Subversion-Arbeitskopien solide und komfortabel aufwertet. Ein Upgrade des Subversion-Clients genügt, um hiervon zu profitieren – kein Server-Upgrade zwingend nötig.

## Server Cache

Serverseitig bietet Subversion 1.7 einen neuen Cache für sowohl Volltexte einzelner Datei-Revisionen, die ansonsten aus den im Repository abgelegten inkrementell aufgebauten Deltas errechnet werden müssen, als auch für einzelne Deltas selbst. Mit einer per Repository abgestimmten Cachegröße lässt sich so einiges an Performance herausholen; das Optimum liegt bei schätzungsweise zwei- bis dreimal der Größe des `trunk` [5]. Bei besonders schneller Netzwerkverbindung kann es sich auch lohnen, Subversions altbekannte Netzwerk-Kompression komplett auszuschalten, um die CPU zu entlasten. Denn ab einer bestimmten Übertragungsgeschwindigkeit wird die Datenkompression zum Flaschenhals des Repository-Zugriffs – und somit überflüssig. Mit der richtigen Einstellung lässt sich also erreichen, dass wiederholt heruntergeladene Inhalte praktisch ohne Aufwand direkt aus dem Arbeitsspeicher ins Netzwerk gespeist werden, so dass in günstigen Fällen Gigabit-Bandbreiten zur Geltung kommen können.

Repository Cache / Netzwerk-Kompression

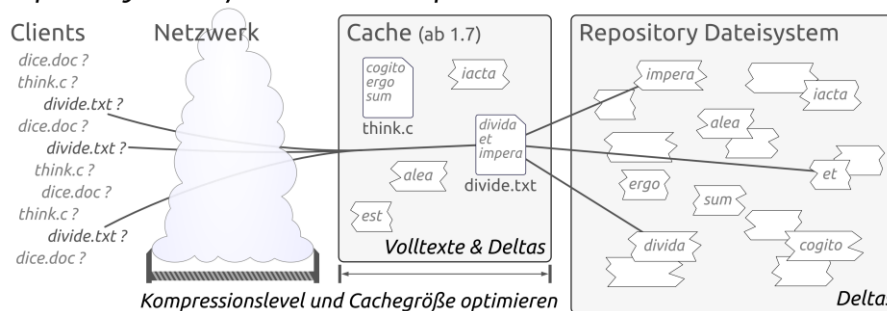


Abbildung 3. Neue Optimierungsmöglichkeiten am Repository

## Fazit

Subversion 1.7 ist vor allem für Power-User und Fortgeschrittene ein sehr spannendes Release, mit dem Subversion sich sozusagen vom letzten Stück CVS-Erbe trennt, den Working Copy Layer gründlich aufräumt und sich für zukünftige Entwicklungen günstig positioniert. Zwar wird bestimmten Nutzern ein bisher "zufällig" verfügbarer Use Case in 1.7 fehlen, nämlich die problemlose Abtrennung von Unterbäumen einer Arbeitskopie, jedoch wird mit *wc-ng* eine solide Basis für weitere Optimierungen geschaffen, die genau diesen Use Case in Zukunft weitaus besser handhaben werden – leider eben nur noch nicht in 1.7. Mit den zahlreich genannten neuen Features, handfesten Optimierungen und Reparaturen an bestimmten Randfällen (nicht zuletzt an File Externals) wird das neue Subversion Release vielen Nutzern Freude bereiten und könnte sogar Entwicklungsprozessen den einen oder anderen ersehnten Use Case erschließen.

Hinweis: Sollten Sie in Ihrem Betrieb auf Subversion angewiesen sein, bietet es sich dringend an, jetzt und bei zukünftigen 1.x.0 Releases, bereits vor dem Release deren Benutzbarkeit für Ihre Prozesse gründlich zu testen, da eine 1.x.0 Release die API relativ streng fixiert. Zum Testen gibt es alpha Releases, für die auch fertig kompilierte Pakete für alle gängigen Plattformen zur Verfügung gestellt werden sollen. So können eventuelle Show-Stopper frühzeitig erkannt und oft bereits vor der endgültigen Fixierung der API korrigiert werden. Erkundigen Sie sich, wie immer, auf <http://subversion.apache.org> [6] oder auf der users@ Mailing-Liste [7].

## Über den Autor



Neels Hofmeyr studiert Technische Informatik an der Technischen Universität Berlin und der University of Stellenbosch, Südafrika. Er ist seit Mitte 2008 aktiv an der Entwicklung von Subversion beteiligt.

Die elego Software Solutions GmbH unterstützt aktuell mit vier Comittern die Weiterentwicklung von Subversion. elego bietet Subversion-Support für den europäischen Raum. D.h. Unternehmen können professionelle Unterstützung bei Installation, Migration und Integration, sowie der Administration und des Betriebs von Subversion in Anspruch nehmen.

## Referenzen

- [1] Subversion ist ein eingetragenes Markenzeichen der Apache Software Foundation (<http://subversion.apache.org>, <http://apache.org>)
- [2] Subversion 1.7 Release Notes: <http://subversion.apache.org/docs/release-notes/1.7.html>
- [3] Automatisierte Benchmarks, die speziell den 1.7 Working Copy mit 1.6 vergleichen, schicken ihre Ergebnisse wöchentlich an die Subversion Entwicklungs-Mailingliste. Eine dieser Mails findet sich unter <http://svn.haxx.se/dev/archive-2011-06/0354.shtml>. Im gleichen Archiv finden sich auch mehrere Diskussionen der Ergebnisse, u.a. <http://svn.haxx.se/dev/archive-2011-04/index.shtml#208>
- [4] Illustration eines *http-roundtrips*: bisher: "Ist x eine Datei?" - "Ja." - "Lösche Datei x." - "Gelöscht." - mit HTTPv2 wird das zu: "Lösche Datei x." - "Gelöscht."
- [5] Server Performance Tuning in den 1.7 Release Notes: <http://subversion.apache.org/docs/release-notes/1.7.html#server-performance-tuning>
- [6] Mitteilung zum 1.7.0-alpha1 Release: <http://subversion.apache.org/news.html#news-20110610>
- [7] Subversion Mailing-Listen: <http://subversion.apache.org/mailling-lists.html>