Ringvorlesung: Themen der Softwareentwicklung

SVN – Nutzung und Managment am Institut

Jan-Peter Bell

Wie verwaltet man komplexe Quelltexte ohne daß man mal vor dem Nichts steht???

Versionsverwaltungssysteme – Leistungen und Ziele

Was wird verwaltet?

Gewöhnliche Dateien – Text-Files, in der Regel keine Binärdaten
z.B.: Quelltexte, Texte, Makefiles, Konfigurationsfiles,...

Mehrere Versionen werden aufbegehoben und bei Bedarf wieder erzeugt.

Ziele von Versionsverwaltungssystemen:

- Parallelen Zugriff organisieren – Synchronisation
- Kontrollierten Zugriff ermöglichen
- Dokumentation der Änderungen
- Kontinuierliche Generationsfolge sichern
- Frühere Versionen wiederherstellen
- Zeit sparen
- Speicherplatz sparen
- Kosten reduzieren
- Sichterheit

Der Vortrag

1. Allgemeines
2. SVN
3. SVN aus der Sicht der Nutzer
4. SVN aus der Sicht des Administrators

```
1. Allgemeines
==============

Historisches - Versionsverwaltungssysteme sind nichts Neues
-----------------------------------------------------------
  SCCS - Source Code Control System
         1972 Marc J. Rochkim, Bell-Laboratorien
         1973 in die UNIX-System eingefügt, kostenpflichtige Software,
         keine frei verfügbaren Quellen, beruht auf 13 Einzelkommandos

  RCS  - Revision Control System
         1983 von Walter F. Tichy, Purdue Universität in West Lafayette
         mehrer Versionen mit signifikanten Unterschieden,
         frei verfügbare Quellen, beruht auf 10 Einzelkommandos

  CVS  - Concurrent Versions System
         1986 Dick Grune (Ableitung von RCS, erster Entwurf)
         1989 Brian Berliner (programmiert), frei verfügbar,
         ein Einzelkommando mit vielen Optionen, netzwerkfähig

  SVN  - Subversion (Next Generation Open Source Version Control)
         Erste Ideen 2000. Abgeleitet von CVS.
         Beseitigt Unzulänglichkeiten von CVS:
               Direktory Versionierung, bessere Fileverwaltung (History)
               Metadatenverwaltung, Verbesserter Netzwerkzugriffe
                        (Apache, SSH, separater Server)
               gut strukturiert, gut verwaltbar, sicherer als CVS.
        Zentrales Repository/Server!!!
        Besteht aus einem Nutzerkommando und mehreren Administrations-
        kommandos, Anbindung an Eclipse, Windows-Clienten, ....
```



```
Die Zukunft
-----------

    GIT - the fast version control system

          Ablösung von BitKeeper
          Ursprünglicher Enwurf von Linux Torvalds zur Enwicklung des Kernels

          für sehr große, komplexe Quelltextbäume
                     z.B.   Linux Kernel,
          Nicht-lineare, verteilte Entwicklung (branching und merging)
          kein zentraler Server
          lokale Repositories
          Datentransfer zwischen Repositories
          Kryptographische Sicherheit der Projektgeschichte
          Gelöschte Daten bleiben vorhanden
          Verbindungen zu CVS , SVN, ARCH
```

```
Versuch einer Einordnung
------------------------

  Zentrale Versionsverwaltungssysteme:

      Open Source
         CVS
         SVN

      Propitäre Systeme
         Alienbrain
         Perforce
         Team Foundation Server
         Visual SourceSafe

  Verteilte Versionsverwaltungssysteme:

      Open Source:
         GNUarch  viele Kommandos, atomare Commits
         Bazaar   Abspaltung von arch (Ubuntu, MySQL, Inkscape, Emacs)
         Darcs    basiert auf Patchen, kein Baum von Revisionen
         GIT      (Linux-Kernel, ...)
         Mercurial (Firefox, Thunderbird, Google, OpenSolaris,
                   Linux-Kernel inoffiziell)
         Monotone  Ideenspender für Git (Pidgin)

      Propitäre Systeme
         BitKeeper (Linux-Kernel - alt - bis 2005)
         ClearCase
```



```
Arbeitsweise von Versionsverwaltungssystemen
--------------------------------------------

    Versionsverwaltungssysteme benutzen in der Regel ein Repository als
    Speicher für die Daten. Der Nutzer agiert grundsätzlich in einer
    Sandbox, die eine Kopie des Repositories darstellt. Die Verwaltungs-
    systeme organisieren den Datentransport zwischen Repository und Sandbox.

    Erzeugen eines leeren Repositories auf dem Server (Administrator)
         init

    Füllen des Repositories mit einem Anfangszustand (Nutzer)
         import

    Füllen einer Sandbox aus einem vorhandenen Repository durch den Nutzer
         checkout

    Übermitteln der veränderten Daten einer Sandbox des Nutzer an das
    bestehende Repository (Nutzer)
         commit

    Hinzufügen von neuen Dateien/Direktories zu einem bestehendes Repository
    (Nutzer)
         import

    Aktualisieren der Sandbox eines Nutzers
         update
```

```
2. SVN - Subversion (Next Generation Open Source Version Control)
=================================================================

Informationsquellen
-------------------

Internet

  http://subversion.apache.org/ - Homepage von Subversion
  http://svnbook.red-bean.com/  - Buch
                                  Version Control with Subversion, OŽReilly
  http://svnbook.red-bean.com/nightly/de/svn-book.html
                                - Buch in deutsch

Buch
  Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato
  Version Control with Subversion, 2. Auflage
  O'Reilly 2008, ISBN 0-596-510330
```



```
Struktur von SVN
----------------
                    +--------------------------------+
                    |Client App (Kommandozeile, GUI)|
                    +--------------------------------+
                                   |
                                   V
                +----------------------------------------+     +---------+
                |SVN-Client-Library  (Sandbox Managment)|--->|Sandbox |
                +----------------------------------------+     +---------+
                                   |
                                   V
                     +------------------------------+
                     |   Repository-Access-Client   |
                     |  DAV    SVNserver     Local  |
                     +------------------------------+
       Network          |           |             |
                        V           V             |
                     +-------+ +----------+       |
                     |Apache| |SVNserver|       |
                     | DAV  | +----------+       |
                     +-------+      |             |
                        |           |             |
                        V           V             V
                    +-------------------------------+        +-----------+
                    |Subversion Repository (DB, FS)|-------->|Repository|
                    +-------------------------------+        +-----------+
```

```
Zugriffsmöglichkeiten auf ein Repository
----------------------------------------

    file:///pfad        - lokales Filesystem
                          Authentifizierung: Zugriffsrechte für Files
                                             lokale Accounts

    http://host/pfad    - Zugriff über http und WebDAV
                          über Apache-Server
                          Authentifizierung: Apache (LDAP, http-passwd)

    https://host/pfad   - Zugriff über http und WebDAV
                          über Apache-Server aber mit SSL
                          Authentifizierung: Apache (LDAP, http-passwd)

    svn://host/pfad     - Zugriff über svnserve-Protokoll
                          Authentifizierung: svnserv

    svn+ssh://host/pfad - Zugriff über ssh- und svn,
                          wie svn aber über ssh-Tunnel
                          Authentifizierung: ssh-Account

Achtung!!! Passwordspeicherung bei http-, https-, svnserv- und file-
                         Authentifizierung
                         ~/.subversion/auth/svn..../1234....
```



```
Subversion Kommandos - Übersicht
--------------------------------
 Kommandos des Nutzers

   svn
           Tool zur Verwaltung der Sandbox und zur Kommunikation
           mit dem Repository (Standardwerkzeug des Anwenders)

   svnversion
           Tool zum Erzeugen einer Versionsnummer einer Sandbox


 Kommandos des Administrators

   svnadmin
           Tool zur Administration von SVN-Repositories (Standardwerkzeug
           des SVN-Administrators auf dem SVN-Server)
   svnsync
           Tool zum Synchronisieren eines Repositories über ein Netzwerk.
   svnlook
           Administrations-Tool zur Inspektion eines SVN-Repository
   svndumpfilter
           Administrations-Tool zum Filtern eines SVN-Dumpfiles
   mod-dav_svn
           SVN-Modul für Apache-Server
   mod-authz_svn
           Authentivizierungs-Modul für Apache-Server für SVN
   svnserve
           Standalone Server, auch mit SSH nutzbar
```

```
Die SVN-Hilfen
--------------

    man svn
    svn help
    svn help <subcommando>
    svnadmin help
    svnadmin help <subcommando>
    svnsync help
    svnsync help <subcommando>

> svn help
Aufruf: svn <Unterbefehl> [Optionen] [Parameter]
Subversion-Kommandozeilenclient, Version 1.6.17.
Geben Sie »svn help <Unterbefehl>« ein, um Hilfe zu einem Unterbefehl
          zu erhalten.
Geben Sie »svn --version« ein, um die Programmversion und die Zugriffsmodule
     oder »svn --version --quiet«, um nur die Versionsnummer zu sehen.

Die meisten Unterbefehle akzeptieren Datei- und/oder Verzeichnisparameter,
wobei die Verzeichnisse rekursiv durchlaufen werden. Wenn keine Parameter
angegeben werden, durchläuft der Befehl das aktuelle Verzeichnis rekursiv.

Verfügbare Unterbefehle:
   add
     ....
   update (up)

Subversion ist ein Programm zur Versionskontrolle.
Für weitere Informationen, siehe: http://subversion.tigris.org/
```



```
3. SVN aus der Sicht des Nutzers
================================

Das SVN-Kommando
----------------

svn UNTERBEFEHL [OPTION] [Parameter]

Das Kommando  »svn«  dient sowohl der Verwaltung der Sandbox als auch
der Verwaltung des Repositories als auch der Kommunikation/Datenübertragung
zwischen Sandbox und Repository durch den Nutzer. Die konkrete Funktion
wird durch den jeweiligen Unterbefehl spezifiziert.

    UNTERBEFEHL -  siehe svn help
                    add       blame    cat        changelist
                    checkout  cleanup  commit     copy
                    delete    diff     export     help
                    import    info     list       lock
                    log       merge    mergeinfo  mkdir
                    move      propdel  propedit   propget
                    proplist  propset  resolve    resolved
                    revert    status   switch     unlock
                    update

    OPTION  - globale Optionen (s.o.)
              locale Optionen - entsprechend dem Unterbefehl

    PARAMETER  - Parameter entsprechend dem Unterbefehl
```

```
SVN-Unterbefehle
----------------

help
----
Ausgabe der Beschreibung der Funktion von SVN einschließlich der
Unterbefehle und ihrer Parameter eines Kommandos.

svn help
    Ausgabe einer Liste der zugelassenen Unterbefehle und ihrer
    Abkürzungen.

svn help help
    Ausgabe aller globalen Optionen (für alle Sub-Kommandos gültig):
      --username PAR      : Benutzername PAR angeben
      --password PAR      : Passwort PAR angeben
      --no-auth-cache     : Anmeldeinformationen nicht zwischenspeichern
      --non-interactive   : keine interaktiven Rückfragen ausgeben
      --trust-server-cert : akzeptiere unbekannte SSL-Server-Zertifikate
                            ohne Nachfrage (aber nur mit »--non-interactive«)
      --config-dir PAR    : Benutzerkonfigurationsdateien aus dem Verzeichnis
                            PAR lesen
      --config-option PAR : Setze Benutzerkonfigurationsoption im Format:
                              DATEI:ABSCHNITT:OPTION=[WERT]
                            Zum Beispiel:
                              servers:global:http-library=serf
svn help UNTERBEFEHL
    Ausgabe der Funktionsbeschreibung, des Aufrufs, der Parameter und
    der zulässigen Optionen für den Unterbefehl.
```



```
checkout (co)
-------------

Checkt eine Arbeitskopie (Sandbox) aus einem Repository (Repository) aus.

svn checkout URL[@REV]... [PFAD]

Falls angegeben, legt REV fest, in welcher Revision die URL zuerst
nachgeschlagen wird.

Ohne Angabe von PFAD wird der Basisname der URL als Ziel verwendet.
Sind mehrere URLs angegeben, wird jede in ein Unterverzeichnis von
PFAD ausgecheckt, dessen Name der Basisname der URL ist.

Bei Angabe von --force lassen nicht versionierte, behindernde Objekte
in der Arbeitskopie den Vorgang nicht automatisch fehlschlagen. Falls
der behindernde Pfad vom gleichen Typ (Datei oder Verzeichnis) wie der
entsprechende Pfad im Repository ist, wird er versioniert, behält
aber seinen ursprünglichen Inhalt. Das bedeutet, dass nicht versionierte
Unterverzeichnisse eines behindernden Verzeichnisses ebenfalls versioniert
werden können. Bei Dateien werden alle Unterschiede im Inhalt wie lokale
Modifikationen in der Arbeitskopie behandelt. Sämtliche Eigenschaften
aus dem Repository werden auf den behindernden Pfad angewendet.
```

```
Optionen:

  -r [--revision] PAR       : PAR (manche Befehle akzeptieren auch einen
                              Wertebereich PAR1:PAR2)
                              Ein Revisionsparameter kann sein:
                                 NUMMER       Revisionsnummer
                                 »{«DATUM»}«  Revision zum Startdatum
                                 »HEAD«       neueste Revision im Repository
                                 »BASE«       Basisrevision der Arbeitskopie
                                 »COMMITTED«  letzte übertragene Revision zu
                                              oder vor BASE
                                 »PREV«       letzte Revision vor COMMITTED
  -q [--quiet]              : nichts oder nur Zusammenfassungen ausgeben
  -N [--non-recursive]      : veraltet; versuchen Sie --depth=files oder
                              --depth=immediates
  --depth PAR               : begrenzt Operation durch Tiefe PAR (»empty«,
                             »files«, »immediates« oder »infinity«)
  --force                   : Durchführung des Befehls erzwingen
  --ignore-externals        : »svn:externals«-Definitionen ignorieren

Globale Optionen:
                             s.o.

Beispiele:

   svn co -r 10 https://svn.informatik.hu-berlin.de/svn/unix-2011
   svn co -r 13 https://svn.informatik.hu-berlin.de/svn/unix-2011
   svn co https://svn.informatik.hu-berlin.de/svn/unix-2011
```



```
commit (ci)
-----------

Überträgt Änderungen einer Arbeitskopie ins Repository.

svn commit [PFAD...]

Eine Logmeldung muss angegeben werden; diese kann jedoch leer sein. Wird
sie nicht mittels einer --message- oder --file-Option übergeben, wird ein
Editor gestartet.
Falls Objekte gesperrt sind oder gesperrte Objekte enthalten, werden diese
nach einer erfolgreichen Übertragung entsperrt.

Optionen:
  -q [--quiet]              : nichts oder nur Zusammenfassungen ausgeben
  -N [--non-recursive]      : veraltet; versuchen Sie --depth=files oder
                              --depth=immediates
  --depth PAR               : begrenzt Operation durch Tiefe PAR (»empty«,
                             »files«, »immediates« oder »infinity«)
  --targets PAR             : Inhalt der Datei PAR als zusätzliche Parameter
                              übergeben
  --no-unlock               : Ziele nicht freigeben
  -m [--message] PAR        : PAR als Logmeldung verwenden
  -F [--file] PAR           : Logmeldung aus Datei PAR lesen
  --force-log               : Gültigkeit der Quelle für die Logmeldung
                              erzwingen
  --editor-cmd PAR          : PAR als externen Editor verwenden
  --encoding PAR            : Wert behandeln, als sei er in der
                              Zeichenkodierung PAR
```

```
  --with-revprop PAR       : Revisionseigenschaft PAR in neuer Revision
                             unter Verwendung des Formats name[=Wert] setzen
  --changelist PAR         : nur auf Elementen der Änderungsliste PAR
                             operieren [äquivalent: --cl]
  --keep-changelists       : Änderungslisten nach Übertragung nicht löschen

Globale Optionen:
                           s.o.

Beispiele:

   cd unix-2011/Einleitung
   vi sysconf.c
   svn ci --message "Beispiel 1" sysconf.c
   svn ci --message "Beispiel 1"
   svn commit
```



```
diff (di)
---------
Zeigt die Unterschiede zwischen zwei Revisionen oder Pfaden an.

svn diff [-c M | -r N[:M] [ZIEL[@REV]...]

   Zeigt die Änderungen an ZIELen, wie sie in REV zwischen zwei Revisionen
   sichtbar sind. ZIELe können alle Arbeitskopien oder URLs sein.

   Falls ZIELe Arbeitskopiepfade sind, wird standardmäßig BASE für N und
   die Version der Arbeitskopie für M verwendet; bei URLs muss N angegeben
   werden und M entspricht HEAD.
   Die Option »-c M« ist äquivalent zu »-r N:M«, wobei N = M-1 gilt.
   Die Verwendung von »-c -M« macht es andersherum: »-r M:N« mit N = M-1.

svn diff [-r N[:M]] --old=ZIEL-ALT[@REVALT] [--new=ZIEL-NEU[@REVNEU]] \
                [PFAD...]

   Zeigt die Unterschiede zwischen ZIEL-ALT in REVALT und ZIEL-NEU in REVNEU.
   Wenn PFADe angegeben werden, sind diese relativ zu ZIEL-ALT und ZIEL-NEU,
   und die Ausgabe ist auf Unterschiede zwischen diesen Pfaden beschränkt.
   ZIEL-ALT und ZIEL-NEU können Pfade der Arbeitskopie oder der Art URL[@REV]
   sein. ZIEL-NEU entspricht ZIEL-ALT, falls es nicht angegeben wurde.
   -r N setzt REVALT auf N, -r N:M setzt REVALT auf N und REVNEU auf M.

diff URL-ALT[@REVALT] URL-NEU[@REVNEU]

   Kurzform für »svn diff --old=URLALT[@REVALT] --new=URLNEU[@REVNEU]«.

svn diff  zeigt alle lokalen Änderungen in einer Arbeitskopie an.
```

```
Optionen:
  -r [--revision] PAR      : PAR (manche Befehle akzeptieren auch einen
                             Wertebereich PAR1:PAR2)
                             Ein Revisionsparameter kann sein:
                                NUMMER          Revisionsnummer
                                »{«DATUM»}«     Revision zum Startdatum
                                »HEAD«          neueste Revision im Repository
                                »BASE«          Basisrevision der Arbeitskopie
                                »COMMITTED«     letzte übertragene Revision zu
                                                oder vor BASE
                                »PREV«          letzte Revision vor COMMITTED
  -c [--change] PAR        : Änderung stammt aus Revision PAR (wie
                             -r PAR-1:PAR). Falls PAR negativ ist,
                             gleichbedeutend zu -r PAR:PAR-1
  --old PAR                : PAR als älteres Ziel verwenden
  --new PAR                : PAR als neueres Ziel verwenden
  -N [--non-recursive]     : veraltet; versuchen Sie --depth=files oder
                             --depth=immediates
  --depth PAR              : begrenzt Operation durch Tiefe PAR (»empty«,
                            »files«, »immediates« oder »infinity«)
  --diff-cmd PAR           : PAR als Vergleichsprogramm verwenden
  --no-diff-deleted        : keine Unterschiede für gelöschte Dateien ausgebe
  --notice-ancestry        : beim Berechnen von Differenzen Vorgänger
                             berücksichtigen
  --summarize              : Zeige eine Zusammenfassung der Ergebnisse
  --changelist PAR         : nur auf Elementen der Änderungsliste PAR operier
                             [äquivalent: --cl]
  --force                  : Durchführung des Befehls erzwingen
  --xml                    : Ausgabe in XML
```



```
  -x [--extensions] PAR    : Vorgabe: »-u«. Wenn Subversion ein
                             externes Vergleichsprogramm aufruft, wird PAR nu
                             an das Programm weitergereicht. Wenn Subversion
                             aber sein internes Vergleichsprogramm benutzt
                             oder Annotierungen anzeigt, kann PAR einen der
                             folgenden Werte annehmen:
                                -u (--unified):
                                   Gibt 3 Zeilen Standardkontext aus.
                                -b (--ignore-space-change):
                                   Ignoriert Änderungen in der Anzahl von
                                   Leerzeichen.
                                -w (--ignore-all-space):
                                   Ignoriert sämtliche Leerzeichen.
                                --ignore-eol-style:
                                   Ignoriert Änderungen im Zeilenendestil.
                                -p (--show-c-function):
                                   Zeigt C-Funktionsname in Diff-Ausgabe.

Globale Optionen:
                            s.o.

Beispiele:

     svn diff
     svn diff -r 13
```

```
revert
------

Stellt eine Datei in der Arbeitskopie wieder her (macht die meisten
lokalen Änderungen rückgängig).

svn revert PFAD...

Dieser Unterbefehl erfordert keinen Netzwerkzugriff und löst Konfliktzustände
auf. Er kann jedoch keine gelöschten Verzeichnisse wieder herstellen.

Gültige Optionen:
  --targets PAR            : Inhalt der Datei PAR als zusätzliche Parameter
                             übergeben
  -R [--recursive]         : rekursiv absteigen, das gleiche wie
                             --depth=infinity
  --depth PAR              : begrenzt Operation durch Tiefe PAR (»empty«,
                            »files«, »immediates« oder »infinity«)
  -q [--quiet]             : nichts oder nur Zusammenfassungen ausgeben
  --changelist PAR         : nur auf Elementen der Änderungsliste PAR
                             operieren [äquivalent: --cl]

Globale Optionen:
                              s.o.

Beispiele:

   svn revert sysconf.c
```



```
update (up)
-----------

Aktualisiert die Arbeitskopie mit Änderungen aus dem Repository.

svn update [PFAD...]

Ist keine Revision angegeben, wird die Arbeitskopie auf den aktuellen
Stand der HEAD-Revision gebracht. Ansonsten wird die Arbeitskopie mit der
durch -r angegebenen Revision synchronisiert.
Für jedes aktualisierte Objekt wird eine Zeile mit einem Buchstaben für die
Aktion ausgegeben. Diese haben die folgenden Bedeutungen

    A  Added    - Hinzugefügt
    D  Deleted  - Gelöscht
    U  Updated  - Aktualisiert
    C  Conflict - Konflikt
    G  merGed   - Zusammengeführt

Ein Buchstabe in der ersten Spalte symbolisiert eine Aktualisierung der
Datei, während Aktualisierungen der Dateieigenschaften in der zweiten
Spalte angezeigt werden.
Ein »B« in der dritten Spalte zeigt an, dass die Sperre für die Datei
aufgebrochen oder gestohlen wurde.
  ..........
```

```
Ab Version 1.5 interaktive Konfliktauflösung im Dialog:

  (p)  zurückstellen      - den Konflikt erst später auflösen
  (df) voller Diff        - alle Änderungen in der zusammengeführten Datei
                            anzeigen
  (e)  editieren          - zusammengeführte Datei in einem Editor ändern
  (r)  aufgelöst          - akzeptieren der zusammengeführten Version der
                            Datei
  (mf) volle eigene Datei - die eigene Version der kompletten Datei
                            akzeptieren (ignorieren fremder Änderungen)
  (tf) volle fremde Datei - die fremde Version der kompletten Datei
                            akzeptieren (verlieren eigener Änderungen)
  (l)  starten            - Starten eines externen Programms zur Konflikt-
                            auflösung
  (h)  Hilfe              - diese Liste anzeigen

Gültige Optionen:

  -r [--revision] ARG   : ARG ist eine Revisionsangabe.
                          Ein Revisions Parameter kann sein:
                                NUMBER       Revisionsnummer
                                "{" DATE "}" Revision zum Startdatum
                                "HEAD"       Neueste im Repository
                                "BASE"       Basisrevision der Arbeitskopie
                                "COMMITTED"  Letzte übertragene Revision bei
                                               oder vor BASE
                                "PREV"       Letzte Revision vor COMMITTED
```



```
  -N [--non-recursive]  : Nicht rekursiv hinabsteigen (veraltet)
  --depth PAR           : begrenzt Operation durch Tiefe PAR (»empty«,
                          »files«, »immediates« oder »infinity«)
  --set-depth PAR       : set new working copy depth to ARG ('exclude',
                          'empty', 'files', 'immediates', or 'infinity')
  -q [--quiet]          : So wenig wie möglich ausgeben
  --diff3-cmd arg       : Verwende ARG als Merge Programm
  --force               : Durchführung des Befehls erzwingen
  --ignore-externals    : »svn:externals«-Definitionen ignorieren
  --changelist PAR      : nur auf Elementen der Änderungsliste PAR operieren
                          [äquivalent: --cl]
  --editor-cmd PAR      : PAR als externen Editor verwenden
  --accept PAR          : automatische Konfliktauflösungsaktion angeben
                          (»postpone«, »base«, »mine-conflict«,
                           »theirs-conflict«, »mine-full«, »theirs-full«,
                           »edit«, »launch«)
Globale Optionen:
                           s.o.

Beispiele:

    svn update
    svn update sysconf.c
    svn update -r 13 sysconf.c
```

```
resolve
-------

Auflösen von Konflikten in Arbeitskopiedateien oder -verzeichnissen.

svn resolve --accept=PAR [Optionen] [PFAD...]

Auflösen von Konfliketen für die angegebe Arbeitskopie  PFAD.

Optionen:
  --targets PAR            : Inhalt der Datei PAR als zusätzliche Parameter
                             übergeben
  -R [--recursive]         : rekursiv absteigen, das gleiche wie
                             --depth=infinity
  --depth PAR              : begrenzt Operation durch Tiefe PAR (»empty«,
                            »files«, »immediates« oder »infinity«)
  -q [--quiet]             : nichts oder nur Zusammenfassungen ausgeben
  --accept PAR             : automatische Konfliktauflösungsaktion angeben
                            (»base«, »working«, »mine-conflict«,
                             »theirs-conflict«, »mine-full«, »theirs-full«)

Globale Optionen:
                           s.o.

Beispiele:

   svn resolve --accept working sysconf.c
```



```
import
------

Überträgt eine nicht versionierte Datei oder einen Dateibaum in das
Repository. Auch zur Anfangsinitialiesierung eines Repositories.

svn import [PFAD] URL

Überträgt rekursiv eine Kopie des PFADes zur URL.
Ohne Angabe von PFAD wird ».« angenommen. Übergeordnete Verzeichnisse
werden soweit erforderlich im Repository angelegt.
Falls PFAD ein Verzeichnis ist, wird dessen Inhalt direkt unterhalb der URL
hinzugefügt.
Nichtversionierbare Elemente wie Gerätedateien und Pipes werden ignoriert,
falls --force angegeben wird.

Gültige Optionen:
  -q [--quiet]             : nichts oder nur Zusammenfassungen ausgeben
  -N [--non-recursive]     : veraltet; versuchen Sie --depth=files oder
                             --depth=immediates
  --depth PAR              : begrenzt Operation durch Tiefe PAR (»empty«,
                            »files«, »immediates« oder »infinity«)
  --auto-props             : automatische Eigenschaften einschalten
  --force                  : Durchführung des Befehls erzwingen
  --no-auto-props          : automatische Eigenschaften ausschalten
  -m [--message] PAR       : PAR als Logmeldung verwenden
  -F [--file] PAR          : Logmeldung aus Datei PAR lesen
  --force-log              : Gültigkeit der Quelle für die Logmeldung
                             erzwingen
```

```
  --editor-cmd PAR           : PAR als externen Editor verwenden
  --encoding PAR             : Wert behandeln, als sei er in der
                               Zeichenkodierung PAR
  --with-revprop PAR         : Revisionseigenschaft PAR in neuer Revision
                               unter Verwendung des Formats name[=Wert] setzen
  --no-ignore                : globale »ignore«- und »svn:ignore«-Einstellungen
                               nicht beachten

Globale Optionen:
                             s.o.

Beispiele:

  cd  dir
  svn import . \
       https://svn.informatik.hu-berlin.de/svn/unix-2011/dir -m "Anfang"
```



```
add
---

Stellt Dateien und Verzeichnisse unter Versionskontrolle und
plant sie zur Übertragung ins Repository ein.
Das tatsächliche Hinzufügen findet erst beim nächsten Übertragen statt.

svn add  PFAD

Gültige Optionen:
  --targets PAR              : Inhalt der Datei PAR als zusätzliche Parameter
                               übergeben
  -N [--non-recursive]       : veraltet; versuchen Sie --depth=files oder
                               --depth=immediates
  --depth PAR                : begrenzt Operation durch Tiefe PAR (»empty«,
                              »files«, »immediates« oder »infinity«)
  -q [--quiet]               : nichts oder nur Zusammenfassungen ausgeben
  --force                    : Durchführung des Befehls erzwingen
  --no-ignore                : globale »ignore«- und »svn:ignore«-Einstellungen
                               nicht beachten
  --auto-props               : automatische Eigenschaften einschalten
  --no-auto-props            : automatische Eigenschaften ausschalten
  --parents                  : direkte Eltern hinzufügen

Globale Optionen:
                             s.o.
Beispiele

  svn add sysconf-neu.c
```

```
delete (del, remove, rm)
------------------------

Entfernt Dateien und Verzeichnisse aus der Versionskontrolle.

svn delete PFAD...

   Jeder PFAD wird zum Löschen bei der nächsten Übertragung
   markiert. Dateien und Verzeichnisse, die noch nicht übertragen
   wurden, werden sofort aus der Arbeitskopie entfernt, es sei denn,
   die Option --keep-local wurde angegeben.
   PFAD(e), die nicht versioniert oder geändert sind, bzw.
   entsprechende Einträge enthalten, werden nur gelöscht, wenn
   die Option »--force« angegeben wird.

svn delete URL...

   Jede URL wird mittels einer sofortigen Übertragung aus dem
   Repository entfernt.

Gültige Optionen:
  --force                  : Durchführung des Befehls erzwingen
  -q [--quiet]             : nichts oder nur Zusammenfassungen ausgeben
  --targets PAR            : Inhalt der Datei PAR als zusätzliche Parameter
                             übergeben
  -m [--message] PAR       : PAR als Logmeldung verwenden
  -F [--file] PAR          : Logmeldung aus Datei PAR lesen
  --force-log              : Gültigkeit der Quelle für die Logmeldung
                             erzwingen
```



```
  --editor-cmd PAR         : PAR als externen Editor verwenden
  --encoding PAR           : Wert behandeln, als sei er in der
                             Zeichenkodierung PAR
  --with-revprop PAR       : Revisionseigenschaft PAR in neuer Revision
                             unter Verwendung des Formats name[=Wert] setzen
  --keep-local             : Pfad in Arbeitskopie beibehalten

Globale Optionen:
                          s.o.

Beispiele

  touch foo
  svn add foo
  svn delete --force foo   # mit löschen von foo

  svn touch foo
  svn add foo
  svn revert foo       # ohne löschen von foo
```

```
list (ls)
---------

Zeigt Verzeichniseinträge des Repositories an.

svn list [ZIEL[@REV]...]

Listet jede ZIEL-Datei und die Inhalte jedes ZIEL-Verzeichnisses, wie
sie im Repository existieren, auf. Wenn ZIEL eine Arbeitskopie ist,
wird die entsprechende URL des Repository verwendet. REV bestimmt,
in welcher Revision zuerst nachgeschaut wird.

Standard für ZIEL ist ».«, d.h. die URL vom Repository des
aktuellen Verzeichnisses.

Mit »--verbose« werden die folgenden Felder pro Objekt angezeigt:

    Revisionsnummer der letzten Übertragung
    Autor der letzten Übertragung
    Falls gesperrt, der Buchstabe »O« (»svn info URL« für Details)
    Größe (in Bytes)
    Datum und Zeit der letzten Übertragung

Gültige Optionen:
  -v [--verbose]           : zusätzliche Informationen ausgeben
  -R [--recursive]         : rekursiv absteigen, das gleiche wie
                             --depth=infinity
  --depth PAR              : begrenzt Operation durch Tiefe PAR (»empty«,
                            »files«, »immediates« oder »infinity«)
```



```
  --incremental            : Ausgabe für Verkettung formatieren
  --xml                    : Ausgabe in XML
  -r [--revision] PAR      : PAR (manche Befehle akzeptieren auch einen
                             Wertebereich PAR1:PAR2)
                             Ein Revisionsparameter kann sein:
                                NUMMER       Revisionsnummer
                                »{«DATUM»}«  Revision zum Startdatum
                                »HEAD«       neueste Revision im Repository
                                »BASE«       Basisrevision der Arbeitskopie
                                »COMMITTED«  letzte übertragene Revision zu
                                             oder vor BASE
                                »PREV«       letzte Revision vor COMMITTED

Globale Optionen:
                         s.o.

Beispiele:

      svn ls
      svn ls --verbose
      svn ls --verbose -R
```

```
status (stat, st)
-----------------

Anzeigen des Status eines Objekts in der Sandbox

svn status [PATH...]

Ohne Argumente - nur die lokal modifizierten Objekte anzeigen.
    Kein Zugriff auf das Repository
-q - Zusammenfassung für lokal modifizierte Objekte anzeigen.
-u - Ausgabe der Informationen mit Zugriff auf das Repository
-v - Ausgabe der Informationen für alle Objekte

Es werden maximal sieben Spalten pro Objekt ausgegeben
  1.Spalte: Zustand
      ' ' no modifications
      'A' Added
      'C' Conflicted
      'D' Deleted
      'I' Ignored
      'M' Modified
      'R' Replaced
      'X' an unversioned directory created by an externals definition
      '?' item is not under version control
      '!' item is missing (removed by non-svn command) or incomplete
      '~' versioned item obstructed by some item of a different kind
  2.Spalte: Modifikationszustand
      ' ' no modifications
      'C' Conflicted
      'M' Modified
```



```
  3.Spalte: Lock-Zustand
      ' ' not locked
      'L' locked
  4.Spalte: History-Zustand
      ' ' no history scheduled with commit
      '+' history scheduled with commit
  5.Spalte: Switch-Zustand
      ' ' normal
      'S' the item has a Switched URL relative to the parent
      'X' a versioned file created by an eXternals definition
  6.Spalte: Lock-Token
      (without -u)
      ' ' no lock token
      'K' lock token present
      (with -u)
      ' ' not locked in repository, no lock token
      'K' locked in repository, lock toKen present
      'O' locked in repository, lock token in some Other working copy
      'T' locked in repository, lock token present but sTolen
      'B' not locked in repository, lock token present but Broken
  7.Spalte: Konflikt
      ' ' normal
      'C' tree-Conflicted
      '*' a newer revision exists on the server
      ' ' the working copy is up to date
```

```
Gültige Optionen:
  -u [--show-updates]       : Aktualisierungsinformation ausgeben
  -v [--verbose]            : zusätzliche Informationen ausgeben
  -N [--non-recursive]      : veraltet; versuchen Sie --depth=files oder
                              --depth=immediates
  --depth PAR               : begrenzt Operation durch Tiefe PAR (»empty«,
                             »files«, »immediates« oder »infinity«)
  -q [--quiet]              : nichts oder nur Zusammenfassungen ausgeben
  --no-ignore               : globale »ignore«- und »svn:ignore«-Einstellungen
                              nicht beachten
  --incremental             : Ausgabe für Verkettung formatieren
  --xml                     : Ausgabe in XML
  --ignore-externals        : »svn:externals«-Definitionen ignorieren
  --changelist PAR          : nur auf Elementen der Änderungsliste PAR
                              operieren [äquivalent: --cl]

Globale Optionen:
                          s.o.

Beispiele:
    svn status
    svn status sysconf.c
    svn status -u sysconf.c
    svn status -v sysconf.c
    svn status -uv sysconf.c
```



```
4. SVN aus der Sicht des Administrators
=======================================

Lokale Nutzung von Subversion - lokales Repository bei einem Nutzer
-------------------------------------------------------------------
           (Selbst ist die Frau)

Erzeugen eines leeren Repository für Subversion (lokal)

      mkdir -p /home/bell/Subversion
      chgrp svnuser /home/bell/Subversion                 # *)
      chmod g+ws /home/bell/Subversion                    # *)
      svnadmin create /home/bell/Subversion

Importieren einer Anfangsversion in das SVN-Repository  (lokal)

      #
      # Quelldirectory ist in:  /home/bell/Src/Einleitung
      #
      svn import /home/bell/Src/Einleitung       \
                 file:///home/bell/Subversion/Einleitung -m "Initialzustand"

      # Repository befindet sich in /home/bell/Subversion/Einleitung
      #

  *)  Notwendig, wenn fremde Nutzer auf das Repository
      zugreifen sollen.  UNIX-File-Zugriffsrechte beachten!!!
      Alle Nutzer müssen in der Gruppe svnuser sein!!!!
```

Arbeiten mit einem lokalen Repository

```
      # neue Sandbox erzeugen
      mkdir Sandbox

      # Füllen der Sandbox
      cd Sandbox
      svn checkout file:///home/bell/Subversion/Einleitung
                   ---------------------------------------

      # Inhalt der Sandbox
      ls Einleitung
         Makefile  sysconf2.c  sysconf.c

      # Veränderungen vornehmen und testen
      cd Einleitung
      vi sysconf.c
      make
      ./sysconf

      # Senden der Änderung
      svn commit
      # nur sysconf.c wird übertragen, die eventuell beim Testen
      # neu erzeugten Files kommen nicht ins Repository
```


Entfernte Nutzung von Subversion - Repository bei einem Nutzer
--

Entfernter Zugriff mittels mittels svn+ssh
Repository ist bei einem Nutzer. Der Zugriff erfolgt über SSH-Tunnel mittels svn-Kommandos. Authentifizierung bei SSH.

Erstellen des Repository auf dem Server!!!

```
      mkdir /home/bell/Subversion
      chgrp svnuser /home/bell/Subversion                  # *)
      chmod g+ws /home/bell/Subversion                     # *)
      svnadmin create /home/bell/Subversion
      chmod -R g+ws /home/bell/Subversion                  # *)
```

Übertragen einer Anfangsversion ins Repository (remote)

```
      # Quellen in /home/bell/Src/Einleitung
      cd /home/bell/Src/Einleitung
      svn import .  svn+ssh://hostname/home/bell/Subversion/Einleitung  \
                   -m "Initialzustand"
      # eventuell lokal auf dem Server nachbessern:
      chmod -R g+ws /home/bell/Subversion                  # *)
```

***) Notwendig, wenn fremde Nutzer auf das Repository zugreifen sollen. UNIX-File-Zugriffsrechte beachten!!! Alle Nutzer müssen in der Gruppe svnuser sein!!!!**

```
Arbeiten mit den Daten mittels svn+ssh (remote)

    # Erzeugen eines Buddelkastens
    mkdir Sandbox
    # Ab in den Buddelkasten
    cd Sandbox
    # Spielzeug holen
    svn checkout \
        svn+ssh://hostname/home/bell/Subversion/Einleitung
        ---------------------------------------------------
    # Anschauen was man bekommen hat
    ls Einleitung
    # neues Spielzeug ausprobieren
    cd Einleitung
    # Spielzeug modifizieren und schauen ob es noch funktioniert
    vi sysconf2.c      # ändern der Daten
    make
    # hoffentlich schöneres Spielzeug zurückgeben
    # Senden der Änderung
    svn commit
```



```
Entfernte Nutzung von Subversion - Repository von Apache2 verwaltet
---------------------------------------------------------------------------
Vorbereiten des Apache-Servers (SuSE 11.4)
  Module dav, dav_svn, authz_svn  aktivieren
    /etc/sysconfig/apache2:
                  APACHE_MODULES=" ... dav dav_svn authz_svn ...
  SVN-Location in Konfiguration einfügen
    /etc/apache2/conf.d/subversion.conf:
        <IfModule mod_dav_svn.c>
             <Location /svn>
                 DAV svn
                 # SVNPath /srv/svn/repos/Einleitung
                 SVNParentPath /srv/svn/repos/
                 AuthType Basic
                 AuthName "Authorization SVN"
                 AuthUserFile /srv/svn/passwdfile
                 Require valid-user
             </Location>
        </IfModule>
    Direktory erzeugen:
        mkdir -p /srv/svn/repos
        chown -R wwwrun:www /srv/svn/repos
        chmod -R g+rws /srv/svn/repos
        svnadmin create --fs-type fsfs /srv/svn/repos/Einleitung
        chown -R wwwrun:www /srv/svn/repos/Einleitung

  Erstinitialisierung schon als entfernter Nutzer
     svn import /home/bell/Src/Einleitung    \
              http://hostname/svn/Einleitung/  -m "Initialwert"
```

```
Anzeigen:

      svn ls http://hostname/svn/Einleitung
             ------------------------------
      svn log http://hostname/svn/Einleitung
              ------------------------------

Auschecken, bearbeiten, einchecken:

      svn checkout http://hostname/svn/Einleitung
      cd Einleitung
      vi sysconf.c
      svn commit
      svn log http://hostname/svn/Einleitung

Im Browser unter

       http://hostname/svn/Einleitung/

anzeigbar.
```



```
Betrieb eines SVN-Servers
-------------------------

SVN-Server: Server der viele SVN-Repositories mit unterschiedlichen
            Nutzergruppen und unterschiedlichen Zugriffsrechten beheimatet.

Entscheidungen, die beim Einrichten eines SVN-Servers zu treffen sind:

  Zugriffsmöglichkeiten:
     svnserv : eigenständige Welt, SVN-spezifische Nutzerverwaltung,
               unsichere Datenübertragung
     svn+ssh : Systemspezifische Nutzerverwaltung (ssh), sichere
               Datenübertragung
     http:     unsichere Datenübertragung, eigenständige Nutzerwerwaltung
     https:    sichere Datenübertragung, eigenständige Nutzerverwaltung
               (bei uns)

  Authentifizierung mittels:
     System         -  nur Nutzer, die auf das System Zugriff haben,,
                       haben Zugriff zum SVN
                       möglich bei: ssh, http, https
     Passwortdatei  -  eigene Passwortdatei
                       möglich bei: http, https, svn
     Ldap           -  Nutzer müssen im LDAP bekannt sein, Nutzer müssen
                       keinen Zugriff zum Server haben.
                       möglich bei: http, https, ssh(mit Serverzugriff)
                       (bei uns)
```

```
Verwaltung:

   Verwaltung der Repositories (neue erzeugen, alte streichen)
   Verwaltung der Zugriffsrechte zu den Repositories
   Verwaltung der Nutzer

      automatisch - http-Oberfläche, wenig Aufwand, keine Kontrolle
                    wer was wann macht - rechtliche Probleme!?!
      manuell     - aufwendig, rechtlich sicher
                    (bei uns - viele Scripte)

Strukturierung der Lage der Daten der Projekte

   Möglichkeiten:
      ein Repository für alle Projekte (Zugriffsrechte??)
      pro Struktureinheit ein Repository
      pro Projekt ein Repository

   abhängig von der Verwaltung, eine gewisse Struktur empfiehlt sich
   bei uns entsprechend der Lehrstühle, pro Projekt ein Repository.
```



```
Tägliche Verwaltungsaufgaben

  Nutzerverwaltung
        Passwortfile, LDAP, LDAP mit eigenständigem Bereich für SVN
        bei uns: LDAP

  Zugriffsrechteverwaltung
        je nach benutztem System
        bei uns: auth_svn mit ensprechenden Konfigurationsfiles
        Konfigurationsfile Apache:
           /opt/csw/apache2/etc/dav_svn_VER.authz

              #
              #      Zugriffsregeln fuer Institutsverwaltung   Mueller
              #
              # Definition der Nutzergruppen
              #  Syntax:
              #       <nutzergruppen> = <user>, <user>, ...
              #
              [groups]
              # Repository: /svn/Data/VER/goya
              grp_goya = xyz, meier, schulze
              grpr_goya = abc
              # Repository: /svn/Data/VER/InstRat
              grp_InstRat = ach1, meier
              grpr_InstRat = abc
              #
```

```
#  Definition der Repositories
#    Syntax:
#       [<name des Repositories>:/]
#       <Zugriffsregel>:=  <Nutzername> = <Zugriffsrechte> |
#                         @<Nutzergruppe> = <Zugriffsrechte> |
#                          *
#       <Zugriffsrechte>:=  r | rw
#

[InstDir:/]
@grp_goya = rw
@grpr_goya = r

[InstRat:/]
@grp_InstRat = rw
@grpr_InstRat = r
```



```
Einrichtung von Repositories
       Automatisch
       manuell - Scripte
               Anlegen des Direktories
               svnadmin create --fs-type fsfs REPOSITORY-PFAD
               erzeugen/ändern von /opt/csw/apache2/etc/dav_svn_XXX.authz
               (apache eventuel neu starten)

Löschen von Repositories
       Automatisch
       manuell - Scripte
          dav_svn_XXX.authz modifizieren
          für Mutige:  rm -rf /svn/Data/LST/REPOSITORY

backup
       Automatisch, Scripte
       Kern:
         svnadmin dump REPOSITORY-Pfad -r 1  > /backup/Backup/REP/1
         svnadmin dump REPOSITORY-Pfad -r $VERSION --incremental \
                                         > /backup/Backup/REP/$VERSION
         svnadmin dump REPOSITORY-Pfad > /backup/Backup/REP/REP.$VERSION

restore
       hoffentlich nie notwendig
       Scripte
         svnadmin create --fs-type fsfs REPOSITORY-Pfad
         svnadmin load  REPOSITORY-Pfad <  /backup/Backup/REP/REP.$VERSION
```

synchronisieren

Durch Nutzer - svnsync

```
cd /home/bell/Subversion   # mein neues SVN-Vertzeichnis
mkdir unix-2011     # neues Repository
svnadmin create /home/bell/Subversion/unix-2011
cp /home/bell/Subversion/unix-2011/hooks/pre-revprop-change.tmpl \
        /home/bell/Subversion/unix-2011/hooks/pre-revprop-change
vi  /home/bell/Subversion/unix-2011/hooks/pre-revprop-change
        exit 0

chmod +x  /home/bell/Subversion/unix-2011/hooks/pre-revprop-change
svnsync init file:///home/bell/Subversion/unix-2011 \
      https://svn.informatik.hu-berlin.de/svn/unix-2011
svnsync sync file:///home/bell/Subversion/unix-2011
```


Sonderwünsche bei der Nutzung

```
       manuell einfügen, durch Hooks
          REPOSITORY/hoooks/...
               post-commit
               pre-commit
               post-lock
               pre-lock
               post-unlock
               pre-unlock
               post-revprop-change
               pre-revprop-change
        Hooks sind nach einem   svnadmin create  immer leer
```

Repositories auf Wanderschaft

```
     backup
     löschen
     restore
```