

Subversion – Wykorzystanie i administracja repozytorium

Michał (plucho) Wronka
plucho@gmail.com

14.01.2006

Część I

Odrobina Teorii

- 1 Repozytorium
 - Ogólna koncepcja
 - Modele wersjonowania

Czym jest repozytorium?

- przechowywanie danych
- struktura drzewiasta katalogów i plików
- zapis i odczyt przez wielu użytkowników
- pełna historia zmian od początku repozytorium

Repozytorium inaczej

„If C gives you enough rope to hang yourself,
think of Subversion as of rope storage facility”

Brian W. Fitzpatrick

Wielu użytkowników a repozytorium

- „Lock-Modify-Unlock”
- „Copy-Modify-Merge”

Część II

Korzystanie z repozytorium

2 Podstawy

- Dostęp do repozytorium
- Cykl pracy
- Eksportowanie

3 Operacje zaawansowane

- Konflikty
- Historia zmian
- Ignorowanie plików
- Odzyskiwanie usuniętych elementów

4 Gałęzie rozwojowe, wersjonowanie

- Gałęzie rozwojowe
- Scalanie
- Etykietowanie wersji

Klienty Subversion

svn konsola

RapidSVN widok drzewiasty projektu

TortoiseSVN integracja z Explorerem, tylko Windows

- <http://subversion.tigris.org/>
- <http://rapidsvn.tigris.org/>
- <http://tortoisesvn.tigris.org/>

Protokoły dostępu do repozytorium

`file://` dostęp lokalny

`http://` Apache2 + mod_dav_svn

`https://` Apache2 + mod_dav_svn + SSL

`svn://` svnserve

`svn+ssh://` svnserve + tunel SSH

Pobranie kodu z repozytorium

- `svn checkout (co)`
- `svn co <dostęp> [użytkownik@serwer] <ścieżka>`

Pobranie kodu z repozytorium

- `svn checkout (co)`
- `svn co <dostęp> [użytkownik@serwer] <ścieżka>`
- `svn co svn://10.10.10.1/usr/svnroot/svn/trunk/`
- użytkownik, hasło – identycznie do kont na serwerze

Cykl pracy użytkownika Subversion

- 1 update
- 2 add, delete, copy, move, mkdir
- 3 status, diff, revert
- 4 merge, resolved
- 5 commit

Etap 1 – update

- `svn update (up)`
- `svn up -r 5`
 - A dodanie pliku
 - D usunięcie pliku
 - U modyfikacja pliku
 - C konflikt
 - G sklejenie różnych wersji (konflikt nie wystąpił)

Etap 2 – add, delete, copy, move, mkdir

- `svn add`
- `svn delete` (`del`, `remove`, `rm`)
- `svn copy` (`cp`)
- `svn move` (`mv`, `rename`, `ren`)
- `svn mkdir`
- ręczna edycja zawartości plików

Etap 3 – status, diff, revert

- `svn status (st)`
- `svn st --verbose`
- `svn diff`
- `svn revert`
- powyższe operacje w trybie offline
- `svn st --show-updates (online)`

Etap 4 – merge, resolved

- `svn merge`
- `svn resolved`

Etap 5 – commit

- `svn commit (ci)`
- `svn ci`
- `svn ci -m "Komentarz zmian, których dokonaliśmy"`

Eksportujemy kod

- `svn export [-r REV] URL PATH`
- `svn export [-r REV] PATH1 PATH2`

- 2 Podstawy
 - Dostęp do repozytorium
 - Cykl pracy
 - Eksportowanie
- 3 **Operacje zaawansowane**
 - Konflikty
 - Historia zmian
 - Ignorowanie plików
 - Odzyskiwanie usuniętych elementów
- 4 Gałęzie rozwojowe, wersjonowanie
 - Gałęzie rozwojowe
 - Scalanie
 - Etykietowanie wersji

Powstawanie konfliktów

- `svn up`
- modyfikujemy plik `foo.c`
- `svn ci => Out of date`

- `svn up`
- `C foo.c`
- `foo.c.mine`, `foo.c.rOLDREV`, `foo.c.rNEWREV`

Rozwiązywanie konfliktów

- ręczna edycja pliku
- `svn resolved <plik>`
- skopiowanie jednego z utworzonych
- `svn resolved <plik>`
- `svn revert <plik>`

svn log

- `svn log --revision 5:9 [plik]`
- `svn log -r 9:5`
- `svn log -r 8`
- `svn log --verbose`
- `svn log -r 8 -v`

Historii ciąg dalszy

- `svn diff [plik|katalog]`
- `svn diff --revision 9`
- `svn diff -r 5:9`

- `svn cat <plik>`
- `svn list (ls) [plik|katalog]`
- `svn info [plik|katalog]`

Jak zignorować pliki?

- mechanizm properties
- właściwość – `svn:ignore`
- `svn propset (pset) (ps)`
- `svn ps svn:ignore *.o <plik|katalog>`
- `svn ps svn:ignore -f <lista> <plik|katalog>`
- `svn propdel (pdel) (pd)`
- `svn pd svn:ignore <plik|katalog>`

Modyfikujemy listę plików ignorowanych

- `svn proplist (plist) (pl)`
- `svn pl <plik|katalog>`
- `svn propget (pget) (pg)`
- `svn pg svn:ignore <plik|katalog>`
- `svn propedit (pedit) (pe)`
- `svn pe svn:ignore <plik|katalog>`

Wskrzyszanie plików i katalogów

- zlokalizuj, kiedy plik został usunięty
- `svn log --verbose`
- skopiuj plik sprzed usunięcia
- `svn copy -r 23
svn://10.10.10.1/usr/svnroot/svn/file.c ./file.c`
- `svn commit`
- historia pliku sprzed usunięcia jest dostępna

- 2 Podstawy
 - Dostęp do repozytorium
 - Cykl pracy
 - Eksportowanie

- 3 Operacje zaawansowane
 - Konflikty
 - Historia zmian
 - Ignorowanie plików
 - Odzyskiwanie usuniętych elementów

- 4 Gałęzie rozwojowe, wersjonowanie
 - Gałęzie rozwojowe
 - Scalanie
 - Etykietowanie wersji

Po co gałęzie rozwojowe? Po co wersjonowanie?

- duże zmiany w kodzie
- jednoczesna praca nad kilkoma wersjami
- „zamrażanie” kodu w danej wersji – 1.0, 1.2
- dostęp do wszystkich nazwanych wersji

Jak to zrobiono w Subversion?

- zalecana struktura katalogów w repozytorium
 - tlug/trunk
 - tlug/branches
 - tlug/tags
- kopiowanie całych drzew katalogów
 - zachowanie historii zmian
 - „cheap copy” – bez fizycznej duplikacji plików

Tworzymy nową gałąź

- `svn co svn://10.10.10.1/usr/svnroot/svn/`
- `svn cp trunk branches/my-branch`
- `svn ci -m "Nowa gałąź"`

- `svn cp svn://10.10.10.1/usr/svnroot/svn/trunk
svn://10.10.10.1/usr/svnroot/svn/branches/nowa -m
"Nowa gałąź"`

Zaczynamy pracę na gałęzi

- `svn co`
`svn://10.10.10.1/usr/svnroot/svn/branches/my-branch`
- `svn switch`
`svn://10.10.10.1/usr/svnroot/svn/branches/my-branch`
- historia przed stworzeniem gałęzi
- `svn switch --relocate <current-path> <new-path>`

Jak i po co scalać?

- poprawianie literówek w trunku i gałęziach
- włączenie kompletnego i przetestowanego kodu z gałęzi do głównej linii rozwojowej
- `svn merge -r 343:344`
`svn://10.10.10.1/usr/svnroot/svn/trunk`
- Subversion nie pamięta scaleń – ryzyko podwójnej aplikacji zmiany – Dobre nawyki

Dobre nawyki

- umieszczanie dokładnego opisu scalenia (wersji) w logach
- scalanie „na sucho” – `svn merge --dry-run`

Scalanie całych gałęzi

- `svn log --verbose --stop-on-copy`

Nabroiliśmy, jak to odkręcić?

- scalanie wstecz
- `svn merge -r 303:302 <repository-path>`

Upragnione 1.0.0

- analogicznie do tworzenia gałęzi rozwojowych
- katalog /tags
- ```
svn cp svn://10.10.10.1/usr/svnroot/svn/trunk
svn://10.10.10.1/usr/svnroot/svn/tags/alpha-release
-m 'komentarz'
```
- ```
svn cp my-repo  
svn://10.10.10.1/usr/svnroot/svn/tags/alpha-release
```

Część III

Administracja

- 5 W pigułce
- Zakładanie repozytorium
 - Konfiguracja dostępu
 - Skrypty

Tworzymy nowe repozytorium

- `svnadmin create <nazwa>`
- `--fs-type <fsfs|bdb>`

- `su svnroot`
- `cd /home/svnroot`
- `svnadmin create tlug`

svnserve

```
/etc/services  
svn 3690/tcp #Subversion  
svn 3690/udp #Subversion
```

```
/etc/rc.d$/inetd.conf  
# SVN svnserve  
svn stream tcp nowait svnroot\  
  /usr/bin/svnserve svnserve -i
```

Kontrola dostępu

```
#conf/svnserve.conf
[general]
anon-access = read
auth-access = write
password-db = passwd
realm = TLUG

#conf/passwd.conf
[users]
plucho=haslo_plaintext
```

Hook scripts

hooks/

- post-commit
- post-lock
- post-revprop-change
- post-unlock
- pre-commit
- pre-lock
- pre-revprop-change
- pre-unlock
- start-commit

Część IV

Załączniki

Odnośniki

- <http://svnbook.red-bean.com/>
- <http://subversion.tigris.org/>
- <http://rapidsvn.tigris.org/>
- <http://tortoisesvn.tigris.org/>

Subversion a CVS

- wersjonowanie całych katalogów, nie indywidualnych plików
- prawdziwa historia – zmiany nazw plików i kopiowanie zachowuje historię zmian
- atomowość dostępu do repozytorium
- zróżnicowane sposoby dostępu do repozytorium
- efektywne gałęzie – „hard-link” – projekt nie „tyje” tak szybko