

Poprvé s SVN (Subversion)

Softwarový systém **SVN** je určen k efektivnímu ukládání vývojových řad softwarových projektů. Je tvořen databází, rozhraním pro přístup k této databázi (obojí je serverovou stranou systému) a programy pro přístup k serveru (klientská strana systému). Správce založí pro daný projekt databázi (**repozitář**) a stanoví přístupová práva; oprávnění uživatelé pak z databáze převezmou (**checkout**) aktuální nebo starší revizi projektu, provedou ve své pracovní kopii úpravy a odešlou je (**commit**) zpět do databáze jako novou revizi. Všechny vzestupně číslované revize zůstávají v databázi, změny se ukládají inkrementálně. Databázi lze klást mnoho typů dotazů – např. sledovat autorské popisy revizí (**log**), seznam a obsah souborů v revizích (**list**, **cat**), porovnávat obsah revizí (**status**, **diff**). Je podporována souběžná práce více autorů na tomtéž projektu, lze oddělovat a selektivně slučovat vývojové větve projektu, exportovat z revizí distribuovatelné balíčky atd. Klient přistupuje k databázi pomocí speciálního protokolu **svn** nebo běžným **http**. Klient v Linuxu (program **svn**) se objeví obvykle spolu se serverem instalací příslušného balíčku (Ubuntu: **apt-get install subversion**), pro Windows je populární např. **TortoiseSVN**. Systém je volně dostupný, některé implementace mohou být komerční. Systém je dokumentován volně dostupnou knihou **Version Control with Subversion** (odkaz dole) a nápovědou zabudovanou v každém příkazu, **svn help command**.

Dotazy

Je-li otevřen anonymní přístup, dotazy může klást každý majitel svn klienta se znalostí polohy repozitáře. V opačném případě je třeba uživatelské jméno a heslo. Není-li nastaveno šifrování (default), hesla leží na obou stranách v otevřeném tvaru, i když po síti otevřená netečou; není tedy vhodné, aby byla příliš důvěrná.

Dotaz na **aktuální revizi** (číslo, datum, popis) v repozitáři (server karel.troja.mff.cuni.cz, repozitář SwarmOceans)

```
svn log svn://karel.troja.mff.cuni.cz/SwarmOceans -r HEAD
```

Dotaz na **historii** revizí (default jsou všechny revize, **-r 1:HEAD**, u velkých projektů se člověk nedočká)

```
svn log svn://karel.troja.mff.cuni.cz/SwarmOceans
```

Dotaz na **jména** adresářů a souborů

```
svn list -R svn://karel.troja.mff.cuni.cz/SwarmOceans
```

Dotaz na **obsah** souboru

```
svn cat svn://karel.troja.mff.cuni.cz/SwarmOceans/README
```

Download

Stahovat (**checkout** neboli **co**) lze aktuální i starší revize projektu, buď celé nebo dílčí adresáře. Stáhnutím se vytvoří lokální pracovní kopie (working copy), obsahující kromě dat projektu i interní databázové informace, sloužící systému k udržování přehledu o následně prováděných změnách. Stáhnout lze i samotná data projektu, obvykle pro další distribuci; zpětný commit z takových dat nelze provést.

Vytvoření pracovní kopie aktuální revize v předepsaném adresáři (včetně interních dat v adresáři **.svn**)

```
svn checkout svn://karel.troja.mff.cuni.cz/SwarmOceans SwarmOceans
```

Download adresáře

```
svn checkout svn://karel.troja.mff.cuni.cz/SwarmOceans/Models SO-Models
```

Download starší revize v její původní podobě

```
svn checkout svn://karel.troja.mff.cuni.cz/SwarmOceans@1 SO-rev1
```

Download souborů ze **starší revize** do struktury aktuální revize

```
svn checkout svn://karel.troja.mff.cuni.cz/SwarmOceans -r 1 SO-rev1
```

Export balíčku (bez interních dat v **.svn** adresářích, nelze provést zpětný commit)

```
svn export svn://karel.troja.mff.cuni.cz/SwarmOceans SO-package
```

Úpravy v pracovní kopii

V pracovní kopii lze editovat soubory a přidávat a rušit soubory a adresáře. Změny editovaných souborů systém registruje automaticky, přidat nové položky je třeba do systému výslovně (**add**, **copy**), vyjmát také (**delete**).

```
svn add file/dir/link
```

```
svn delete file/dir/link
```

```
svn copy file1 file2
```

```
svn move file1 file2
```

Obhlídka provedených úprav: v aktuálním adresáři a podadresářích lze sledovat změny pracovní kopie proti výchozímu stavu vyjádřené souhrnnými symboly (**status**) nebo podrobným výpisem (**diff**).

```
svn status
```

 (? nesledovaná položka, tj. bez svn add, A added, C conflicted, D deleted, M modified, R replaced aj.)

```
svn diff; svn diff -r rev1:rev2 file
```

Koordinace pracovní kopie a aktuální revize v repozitáři: **update** aktualizuje pracovní kopii z aktuální revize, **revert** ruší provedené změny a vrátí pracovní kopii do jejího výchozího stavu. Příkaz **resolve** slouží k řešení konfliktů, vzniklých souběžnou prací více uživatelů na týchž souborech; souběžná práce na různých částech projektu ke konfliktům nevede.

```
svn update
svn revert file/dir/link
svn resolve ...
```

Odeslání změn

Odeslat (**commit** neboli **ci**) změny do repozitáře lze jen z aktuální pracovní kopie, tedy z kopie odvozené z revize, která je v repozitáři dosud aktuální. Byl-li mezitím repozitář aktualizován (jiným uživatelem), je třeba nejprve provést **svn update**. Každý commit je vhodné doplnit komentářem vystihujícím provedené změny a dostupným pak příkazem **svn log**. Po commitu se zvýší číslo revize v repozitáři, nikoliv v pracovní kopii; má-li pracovní kopie získat totéž číslo, je vhodné provést ještě jeden **svn update**, jinak je ovšem možné a běžné po commitu pracovní kopii smazat.

```
svn commit -m "Message - description of changes"
svn commit -f message-file
```

Nová data (obvykle první revizi) lze do repozitáře **importovat** hromadně bez vytváření pracovní kopie. Odešle se obsah adresáře, nikoliv samotný adresář.

```
svn import dir svn://server/repository -m "Initial import."
```

Ukázka pracovního cyklu

Checkout (download), přidání dat, ověření provedených změn, ověření aktuálnosti pracovní kopie, commit (upload).

```
svn checkout svn://server/repository my-copy   Checked out revision 111.
cd my-copy                                     (přesun do pracovní kopie)
edit old-file                                  (systém změny registruje sám)
edit new-file                                  (nový soubor nutno do systému přidat následujícím příkazem)
svn add new-file                               (soubor v pracovní kopii registrován jako sledovaný)
svn status                                     A new-file, M old-file
svn update                                     At revision 111. (test aktuálnosti pracovní kopie)
svn commit -m "Added file new-file."          Adding ..; Sending ..; Transmitting ..; Committed revision 112.
svn update                                     At revision 112. (synchronizace čísel revizí v repozitáři a pracovní kopii)
```

Konfigurace

Klient v Linuxu lze konfigurovat editací souboru `~/.subversion/config`. K žadaným nastavením patří např. automatické korekce koncových znaků řádků nebo expanze kódu `Id` při commitu na údaj zahrnující číslo revize, datum, čas a uživatelské jméno.

Administrátorské příkazy

Vytvoření repozitáře na serveru (prázdna databáze s default konfiguračními soubory)

```
svnadmin create /nfs2/svn/repository
```

Konfigurační soubory v `repository/conf`

`svnserve.conf` (globální práva, názvy konfiguračních souborů)

`passwd` (uživatelská jména a hesla)

`authz` (uživatelské skupiny, přístupová práva k adresářům a souborům)

Informace o poslední revizi

```
svnlook info /nfs2/svn/repository
```

Start síťového serveru pro svn protokol

```
svnserve -d -r /nfs2/svn
```

Odkazy

domácí stránka

<http://subversion.apache.org>

kniha **Version Control with Subversion** (HTML, PDF)

<http://svnbook.red-bean.com>

wiki

http://en.wikipedia.org/wiki/Apache_Subversion