
Subversion pro každého

Ladislav Prskavec <ladislav@prskavec.net>

Table of Contents

1. Předmluva	2
2. O knize	2
2.1. Licence	2
3. Úvod	2
3.1. Co je to Subversion?	2
3.2. Historie Subversion	2
3.3. Jak funguje Subversion	3
3.4. Základní pojmy	4
3.5. Doporučená struktura repozitory	5
4. Instalace	6
4.1. Získáme Subversion	6
4.2. Windows	6
4.3. Linux - Debian / Ubuntu	8
5. Začínáme	9
5.1. TortoiseSVN	9
5.2. SVN - řádkový klient	12
5.3. Běžný pracovní postup	13
5.4. Export pro hosting, ftp apod.	13
6. Pokročilejší funkce	14
6.1. Vlastnosti	14
6.2. Vlastnosti projektu	15
6.3. Migrace repozitory	16
6.4. Mirror repository pomocí svnsync	16
6.5. Branches a tags - větve a štítky	17
6.6. Merge v souborech	18
7. Integrace SVN s editory a IDE	19
7.1. Eclipse	19
7.2. Microsoft Visual Studio	20
7.3. NetBeans, Komodo IDE, SciTE a další	21
7.4. JetBrains (RubyMine, PHPStorm, IntelliJ IDEA)	23
8. Distribuované systémy pro správu verzí	24
8.1. Přechod od Subversion k Mercurial	24
8.2. Spolupráce distribuovaných systémů se Subversion	24
9. Nástroje pro práci se Subversion	25
9.1. Commit Monitor	25
9.2. StatSVN	27
9.3. Generování changelogu pomocí svn2cl	28
10. Literatura a použité zdroje	29
10.1. SVN klienti	30
10.2. SVN servery	30
10.3. Dokumentace	30
11. Používané zkratky a anglické příkazy	30

1. Předmluva

Tato kniha se nesnaží být překladem SVN book i když z ní samozřejmě vychází. Motivací pro mě byli jak vlastní zážitky se Subversion, tak pozdější zavádění Subversion ve větším měřítku. Dnes si práci bez Subversion nedovedu představit a kvalitní česká kniha zaměřená převážně na uživatele neexistuje. Snad tato kniha přinese ještě větší rozšíření Subversion.

Pokud kniha obsahuje nějaké skripty tak jsou buď psané ve Windows Vista nebo na Linuxu kde používám distribuci Ubuntu.

Pokud budete mít jakékoli připomínky, náměty nebo byste se chtěli podílet na rozrůstání obsahu knihy, obraťte se na mě nejlepe na email <ladislav@prskavec.net [mailto:ladislav@prskavec.net]> a do předmětu uveďte "SVN kniha".

2. O knize

2.1. Licence

Tento dokument je určen výhradně pro osobní potřebu seznámení se systémem Subversion. Jakékoliv jiné použití, včetně dalšího šíření, poizování kopií apod. je výslovně zakázáno a bude považováno za porušení autorských práv.

3. Úvod

3.1. Co je to Subversion?

Zcela vlastními slovy bych pro znalé řekl, že je to nástroj, který nahradil CVS. Pro ty, kterým tato věta nic neříká je určen další odstavec.

Subversion je nástroj ze skupiny SCM (správa obsahu zdrojových kódů - Source Content Management). Nejvíce rozšířený nástroj v minulých letech a stále i dnes je CVS, který má několik zásadních nevýhod, které donutili CollabNet k tomu, aby se vrhli do vývoje Subversion.

Subversion slouží ke správě zdrojových kódů, které jsou napsány v jakémkoliv jazyce jak programovacím tak lidským. Nativně podporuje kódování UTF-8 a proto není problém psát dokumenty anglicky, rusky nebo svahilsky. SVN uchovává vytvářené revize, to znamená, že uchovává stav dokumentů z doby, kdy jste dali vykonat příkaz commit, který slouží k odeslání zdrojových dat na server. Můžete si pomocí Subversion spravovat verze textového dokumentu jako je například tato kniha psaná v Docbooku a nebo zdrojové kódy v libovolném programovacím jazyce (třeba i vlastním).

Subversion patří k centralizovaným SCM systémům jako je CVS, Perforce, Clearcase a další. Existují také distribuované SCM systémy, které nemají klasickou architekturu client/server. Například Bazaar, Mercurial nebo Git. Porovnání a odkazy na jiné SCM najdete na stránkách Better SCM

3.2. Historie Subversion

Subversion vznikl oficiálně 31. srpna 2001 a verze 1.0 byla k dispozici v roce 2004.

1.1 září 2004 (FSFS)

1.2 květen 2005 (locks)

1.3 prosinec 2005 (svnserve)

1.4 září 2006 (svnsync)

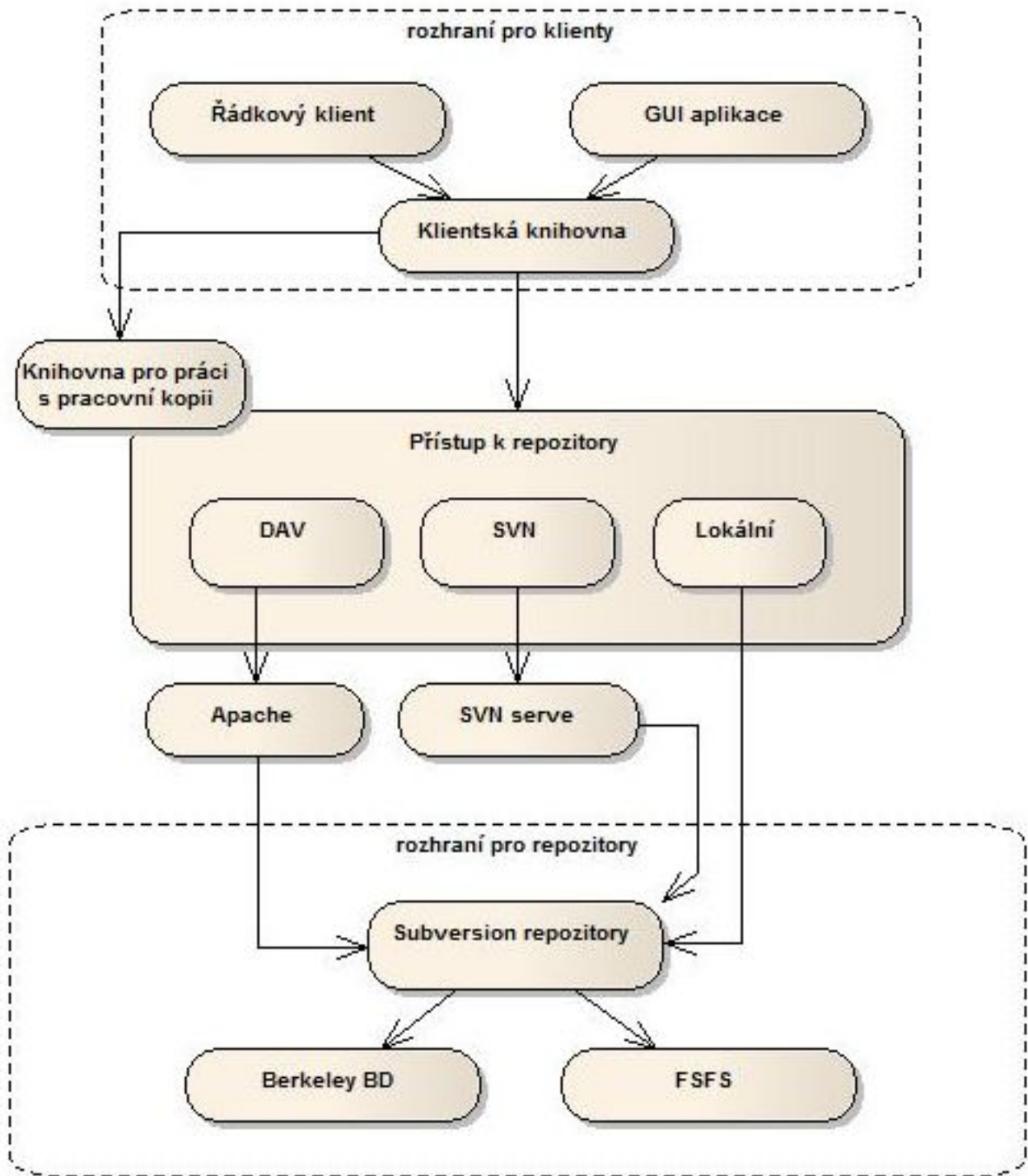
1.5 červen 2008 (merge tracking support)

1.6 březen 2009

Vývoj Subversion neustále pokračuje a přibližuje se jeho kvalita komerčním systémům jako je Perforce nebo Clearcase. V opensource komunitě, ale začíná také získávat popularitu Git, který je narušil od Subversion distribuovaný. Git je například používán v komunitě OpenSUSE nebo při vývoji jádra Linuxu.

3.3. Jak funguje Subversion

Architekturu subeversion nejlépe znázorňuje obrázek v [SVNBOOK] na straně 22.



3.4. Základní pojmy

3.4.1. Repository (repozitář, centrální úložiště)

Umožňuje organizovat projekt a spravovat jeho verze. Fyzicky je uloženo na souborovém systému serveru. K repository se přistupuje přes Repository Access Layer (RA) systému Subversion a jeho správa se provádí klientskými nástroji.

3.4.2. Branch (v#tev)

Slouží k organizaci repository, jedná se o jakousi analogii s adresá#i. Pokud se z repository vyzvedne v#tev, na klientovi vznikne adresá#ová struktura, která p#esn# odpovídá v#tvím v repository.

3.4.3. Revision (revize)

Revize je po#adové #íslo každé zm#ny. Slouží ke sledování zm#n ve v#tvích v #ase. Každá zm#na v n#jaké v#tvi vytvo#í novou revizi v rámci celé repository. Revize obsahuje informace o tom, co bylo zm#n#no, kdo zm#nu provedl, poznámku a #as.

3.4.4. Pracovní kopie (working copy)

Kopie dat z ur#ité v#tve z repository v aktuální revizi na pevný disk lokálního klienta. Do pracovní kopie je možné provád#t zm#ny, které je možné commitem uložit zp#t do repository.

3.4.5. Commit

Odeslání zm#n provedených od posledního commitu do repository. Commit je nej#ast#ji používaná zm#na p#i práci s repository. Pokud se provádí commit celé pracovní kopie, jedná se o atomickou operaci, jsou odeslány veškeré zm#ny ve všech objektech ve správ# verzí; pokud dojde k n#jaké chybě p#i p#enosu, není commit pro ostatní uživatele repository zviditeln#n, není vytvo#ena nová revize.

3.4.6. Konflikt

Konflikt je stav, který signalizuje, že stejný objekt, který má být práv# commitován, byl zm#n#n n#kým jiným a nachází se v repository v aktuální revizi v jiné podob#, než jaký je v pracovní kopii. Nelze provést commit celé pracovní kopie, pokud se v ní nachází jeden nebo více souborů v konfliktu.

3.4.7. Cheap-copy

Technika, kterou se realizují kopie provád#né v rámci repository. Objekty nejsou v repository fyzicky duplikovány, ale jsou vytvo#eny tzv. odkazy (link) na kopírované objekty. Zjednodušen# lze chápat takovýto link jako informaci o URL s #íslem revize. Díky tomu má SVN nízké nároky na datový prostor.

3.5. Doporu#ená struktura repository

Stukturu repository lze jen doporu#it, obvyklá struktura je:

```
/project_name
  /trunk
  /tags
  /branches
```

Stuktura m#že samoz#ejm# obsahovat i další adresá#e nap#. vendors pro kód, t#etí strany nezbytný k projektu. Existují samoz#ejm# další možnosti v literatu#e se uvádí nap#.

```
/trunk
  /project_1
  /project_2
/tags
  /project_1
  /project_2
/branches
  /project_1
  /project_2
```

4. Instalace

4.1. Získáme Subversion

Subversion je k dispozici ve formě zdrojových kódů nebo ve formě binárních balíčků pro různé operační systémy. Subversion je dostupný pro Windows, Mac OS, různé distribuce linuxu i pro některé unixy. Veškerý potřebný software stáhnete na stránkách <http://subversion.apache.org>

4.2. Windows

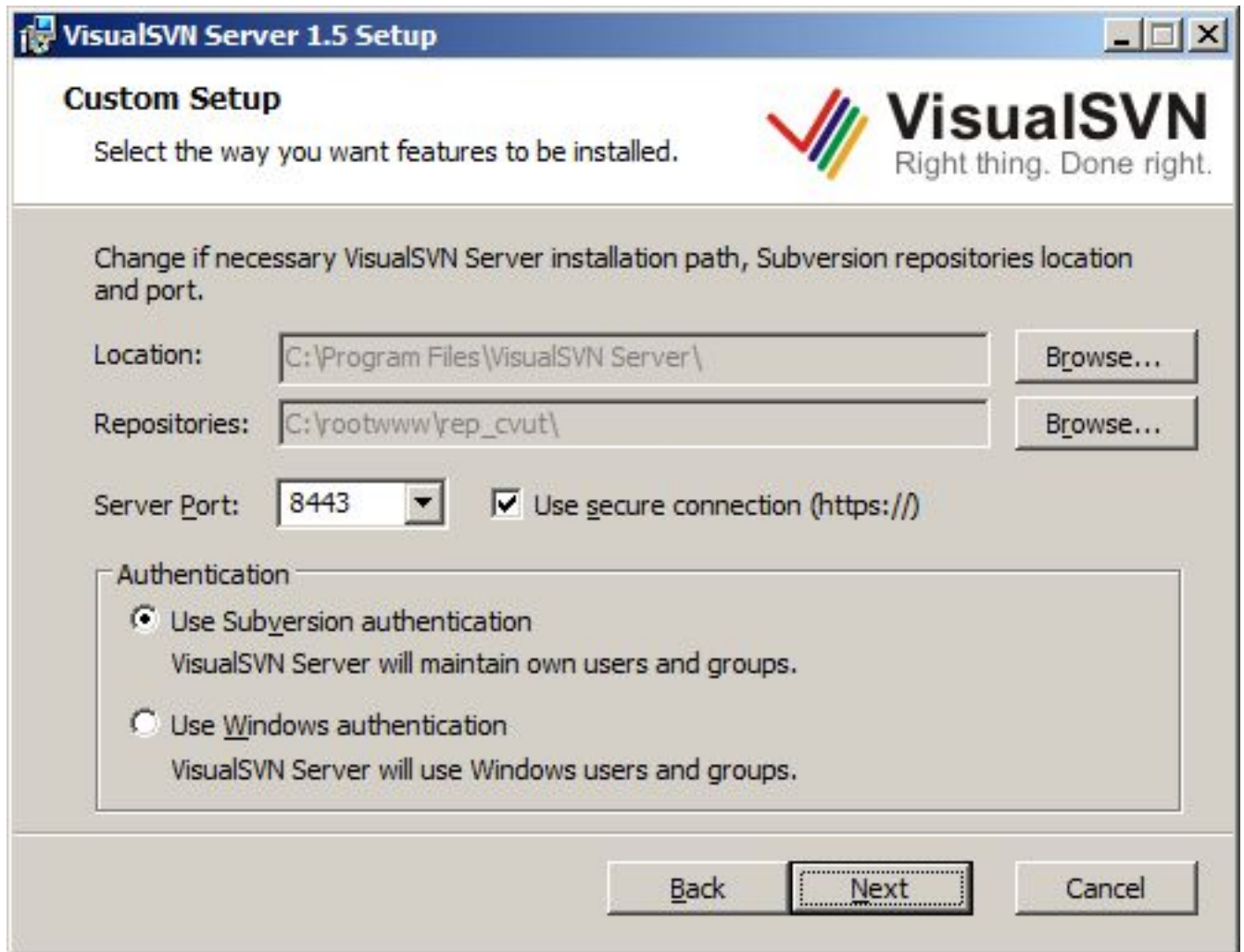
Ve Windows je několik programů, které si můžete pro práci zvolit a to buď klasického řádkového klienta a server s podporou svnserv. Nebo jako klienta použijete TortoiseSVN (TSVN), který umožňuje buď jako klient i server (s omezením na lokální přístup k repozitory). Pokud chcete přistupovat přes http a https doporučil bych k instalaci VisualSVN Server.

Pokud nepotřebujete, aby někdo přistupoval k vašim lokálním repozitory doporučuji TSVN. Tento program umí všechny funkce a má pěkné prostředí, které se integruje do Windows Exploreru. Je nejpokročilejší klient, který umožňuje i tvorbu hooks skriptů na straně klienta.

4.2.1. VisualSVN server

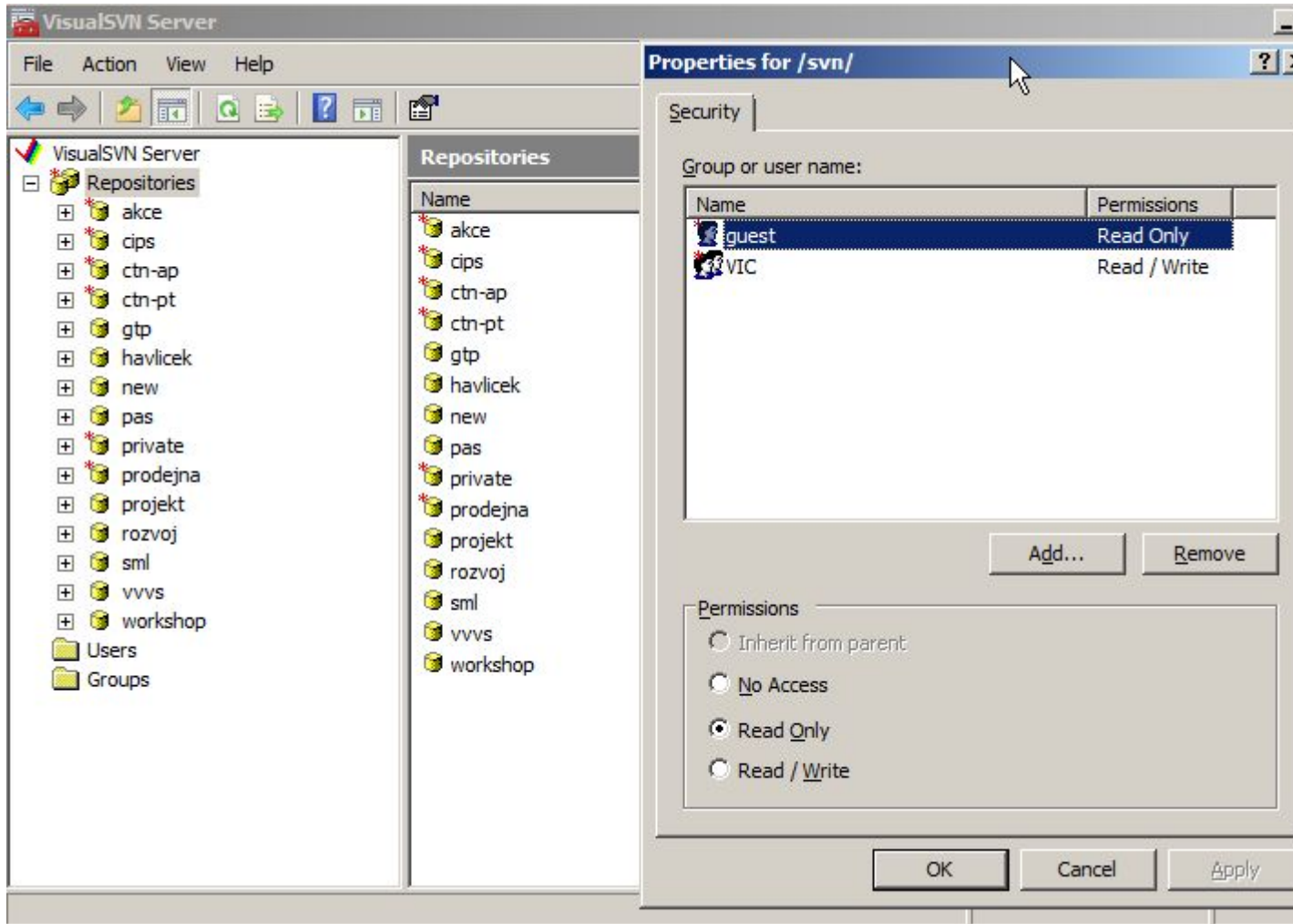
VisualSVN je jen Apache server s mod_dav_svn doplněný administrační konzolí a webovým prohlížečem repozitory.

Instalace je VisualSVN serveru je jednoduchá. Stáhnete si instalátor, který vás pomocí wizarda provede. Můžete navolit kam se server nainstaluje i kde máte adresář s repozitory. Můžete také použít kromě Subversion autentifikace i Windows autentifikaci. Volitelné je i https připojení.



Nastavení práv pro jednotlivé repozitory se dá provést pomocí administrační konzole. VisualSVN také disponuje webovým prohlížečem repozitory, který je dostupný na adrese <https://localhost:8443/> pokud to máte v základním nastavení lokálně, jinak na příslušné adrese a portu.

Pro nastavení práv slouží VisualSVN Server Manager, kde máte vidět jak repozitory, tak uživatele a skupiny. Potom přes pravé tlačítko a volbu Security nastavíte práva buď pro všechna repozitory najednou nebo u každého zvlášť.



4.3. Linux - Debian / Ubuntu

Pro instalaci na Ubuntu použijte zdroje přímo z repozitory.

```
sudo apt-get install subversion
sudo apt-get install subversion-tools
```

Balík subversion obsahuje jak klienta tak server (svnserv). V balíku subversion-tools jsou utility (svn-backup-dumps, svn-clean, svn2cl, mailer apod.)

Já používám svn2cl pro generování changelog#, mailer používám v post-commitech, abych v#d#l co se d#je v repozitá#ích.

Pro přístup lze použít file://, svn:// nebo svn+ssh://. Pokud chcete použít http:// nebo https:// musíte to ud#lat p#es Apache a WebDAV.

```
sudo apt-get install libapache2-svn
```

Konfigurace se potom provede v /etc/apache2/mods-available/dav_svn.conf. Ukázka je s p#epína#em SVNListParentPath, kdy nám dovolí přístup ke všem repozitory, které máme v adresá#i /srv/svn.

```
<Location /svn/>
  DAV svn
```



```

SVNPath /srv/svn
SVNListParentPath On
AuthType Basic
AuthName "Test subversion repository"
AuthUserFile /etc/subversion/passwd
Require valid-user
</Location>

```

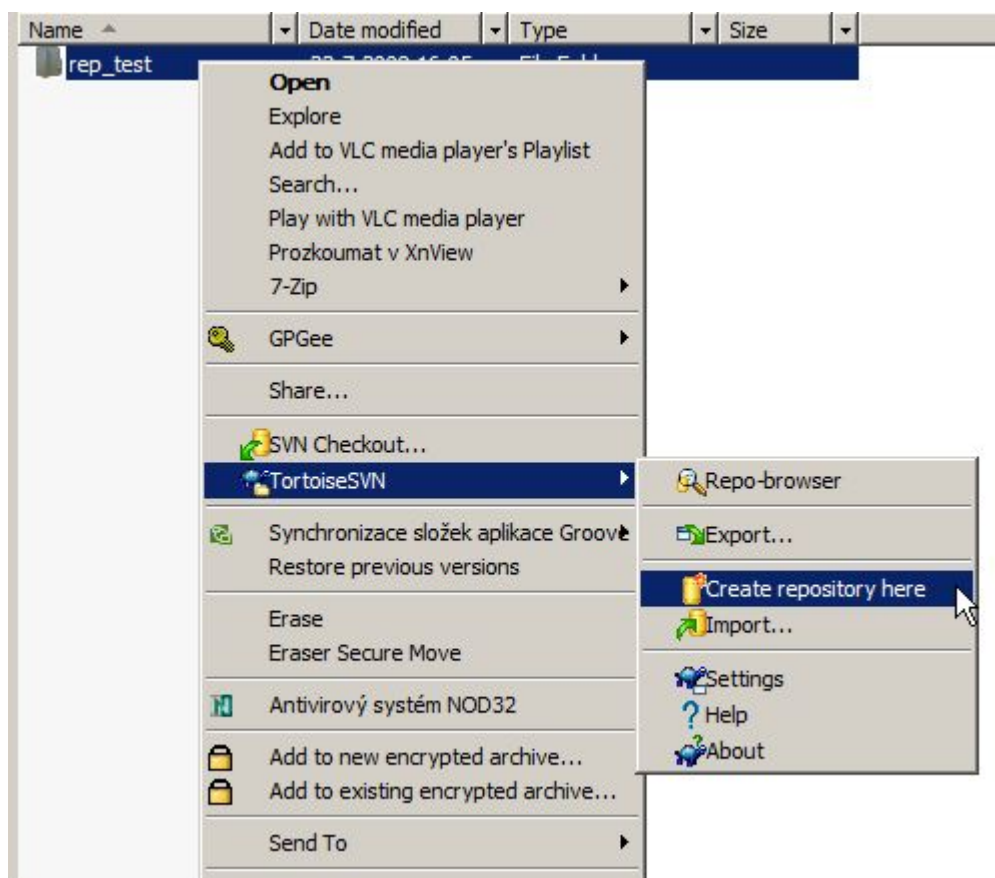
Podrobný návod pro Ubuntu najdete například ve wiki [<https://help.ubuntu.com/community/Subversion>].

5. Začínáme

5.1. TortoiseSVN

5.1.1. Vytváříme repozitory

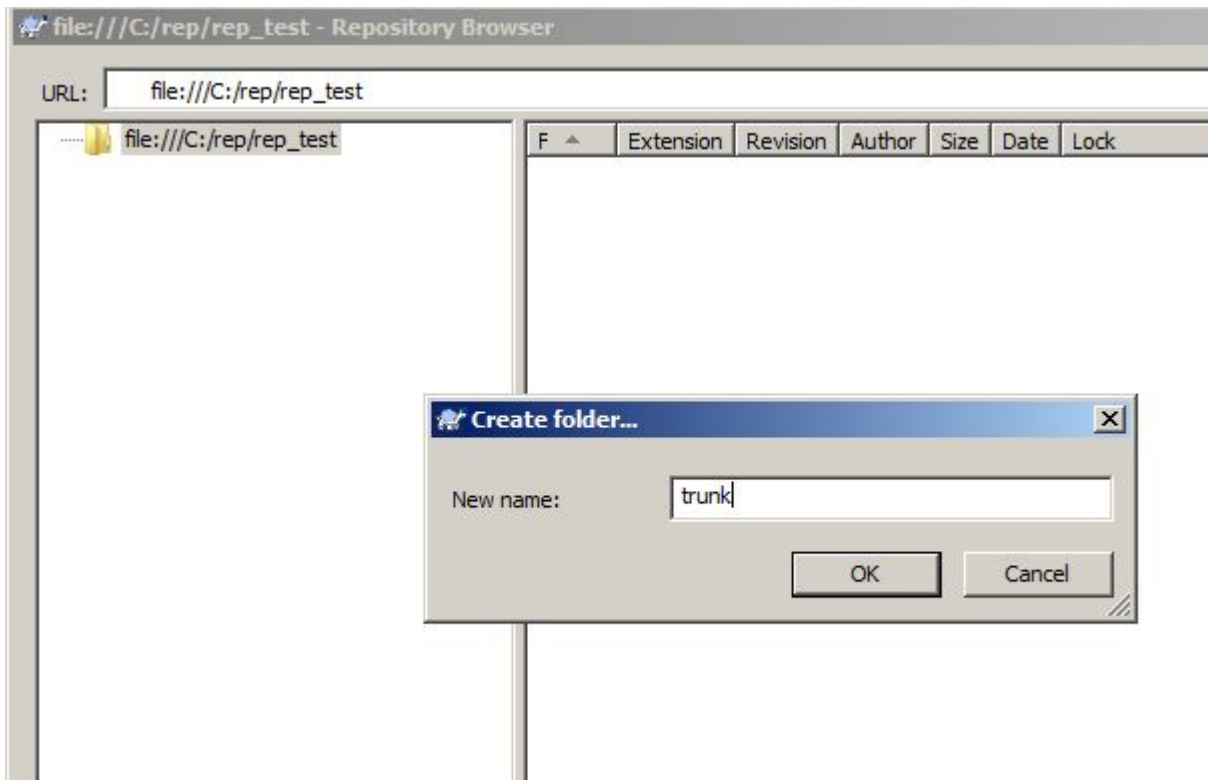
Vytvoříme adresář pro repozitory například `c:/rep` a v něm adresář pro projekt například `rep_test` a na adresáři přes pravé tlačítko se dostaneme do menu TSVN, kde před volbu Vytvořit repozitory zde (Create repository here) vytvoříme repozitory jak ukazuje obrázek. Po úspěšném vytvoření repozitory si můžeme prohlédnout adresář a uvidíme soubory co vytvořil Subversion.



Struktura repozitory je na dalším výpisu. V adresáři `conf/` jsou konfigurační soubory pro přístup k repozitory, v adresáři `db/` jsou uložena data a souboru `current` je aktuální revize. V adresáři `hooks/` jsou šablony pro hooks skripty, které umožní vykonat nějakou akci před určitou akcí v Subversion.

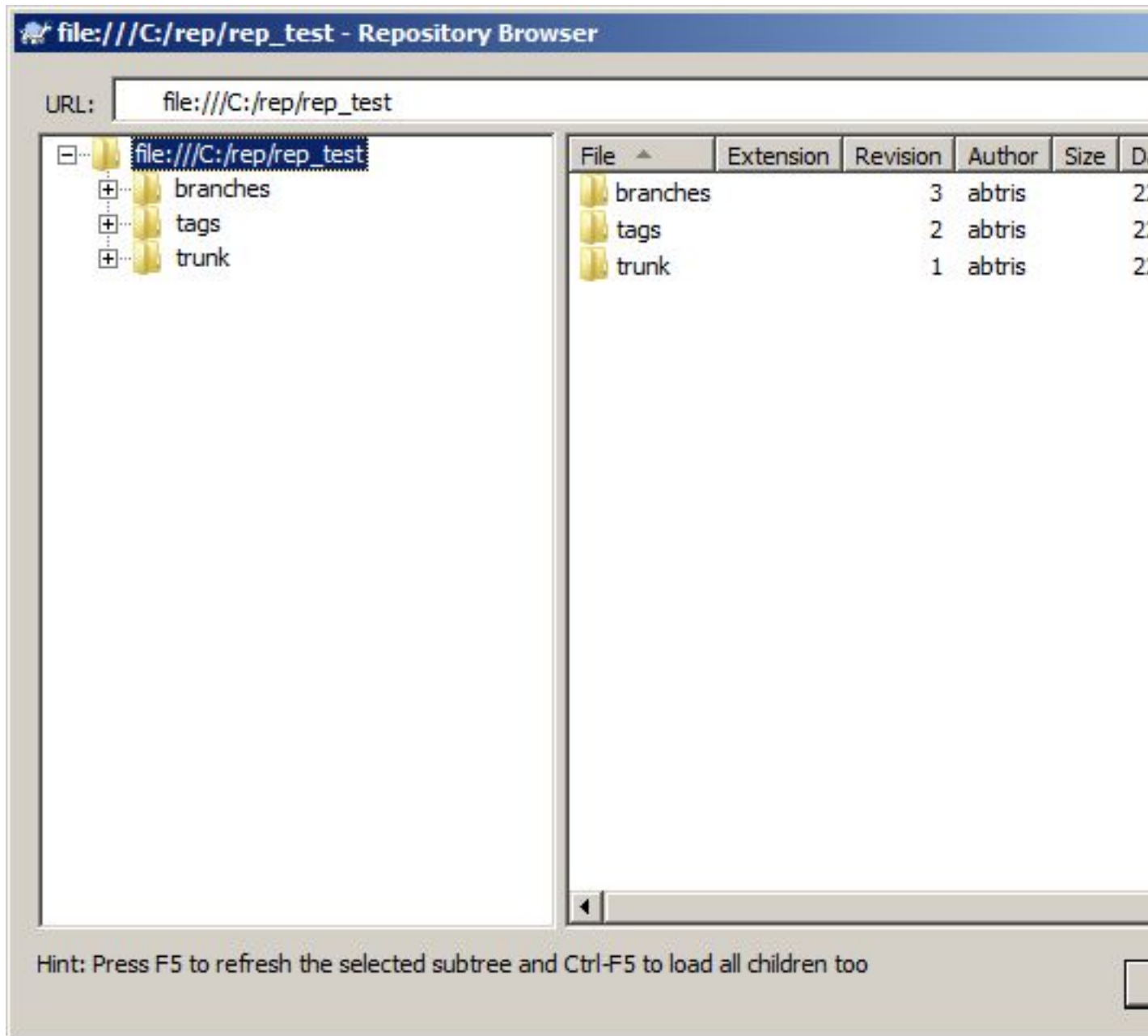
```
conf/  
  authz  
  passwd  
  svnserve.conf  
db/  
  ..  
  current  
  ..  
hooks/  
  post-commit.tmpl  
  post-lock.tmpl  
  post-revprop-change.tmpl  
  post-unlock.tmpl  
  pre-commit.tmpl  
  pre-lock.tmpl  
  pre-revprop-change.tmpl  
  pre-unlock.tmpl  
  start-commit.tmpl  
locks/  
format  
README.txt
```

Po vytvoření repozitáře je dobré otevřít repozitář browser a vytvořit základní strukturu pomocí příkazu vytvořit adresář "create folder".



V repozitáři můžete vytvořit vlastní adresáře, ale doporučuje se nejdříve vytvořit strukturu do které nainportujete svoje adresáře.

Struktura slouží k tomu, abychom vedli hlavní kmen projektu (trunk) a mohli také vést větve (branches) a dát nálepky (tags) pro určité vydané verze.

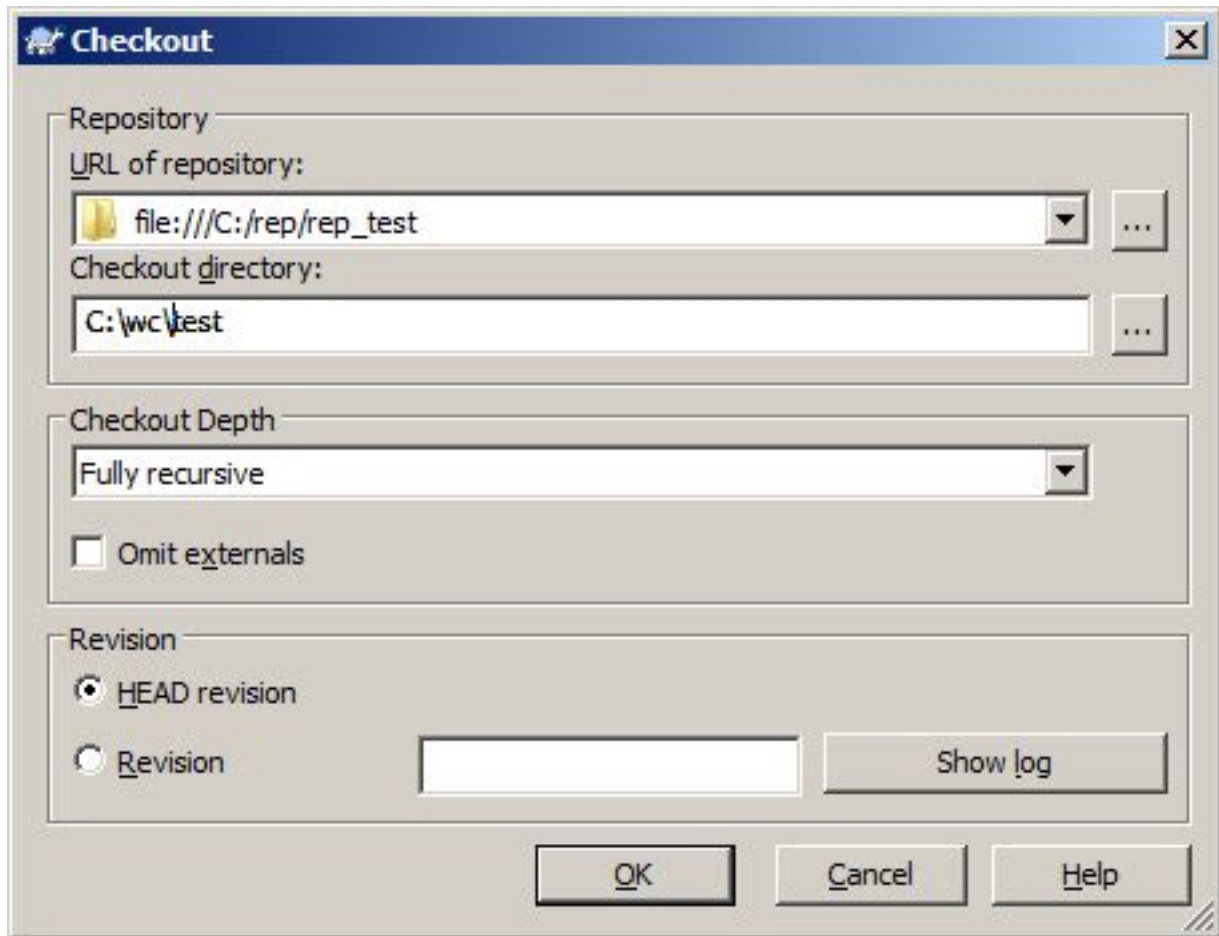


5.1.2. Import zdrojových kódů

Pokud už máme něco co chceme do repozitáře importovat uděláme to pomocí příkazu `import`. Jinak můžeme tento krok přeskočit, protože strukturu můžeme vytvořit pomocí repozitářového prohlížeče.

5.1.3. Vytváříme pracovní kopii

Provedeme `checkout` na repozitář a začneme pracovat.



5.2. SVN - #ádkový klient

Ve stru#nosti zopakují stejné postupy jako ve TSVN.

Vytvo#íme repozitory

```
svnadmin create --fs-type fsfs c:/rep/rep_test
```

Rozdíl mezi Windows a Linuxem je jen ve zp#sobu zadávání cesty k repozitory. Obdobn# to m#žete aplikovat na všechny p#íklady. SVN se ve funk#nosti mezi Windows a Linuxem nijak neliší.

```
svnadmin create --fs-type fsfs /srv/svn/rep_test
```

V #ádkovém klientu nem#žeme vytvo#it adresá#e p#ímo, ale ud#láme je rovnou p#i prvním importu soubor#, tento postup lze použít i p#es TSVN.

```
cd c:/tmp
mkdir test
cd test
mkdir trunk
mkdir tags
mkdir branches
svn import --message "Initial import" file:///C:/rep/rep_test
```

Potom vytvo#íme pracovní kopii

```
svn checkout file:///C:/rep/rep_test/trunk c:/wc/test
```

5.3. Běžný pracovní postup

Nejdříve si aktualizujte svoji pracovní kopii.

```
svn update
```

nebo

```
svn up
```

Skoro každý příkaz má zkratku, která se hodí aby jste to nemuseli rozepisovat. Stačí se podívat na `svn help` a uvidíte je.

Potom vytvoříte nové soubory, smažete staré soubory, změníte obsah. Pomocí příkazu `svn diff` si ověříte můžete zobrazit změny své pracovní kopie proti verzi v repozitory.

```
svn diff
```

Pokud jste pracovali s pracovní kopii a přidávali nebo mazali soubory, je dobré vědět zda jsou již přidány či v jakém stavu jsou to zjistíme pomocí `svn status` a potom můžeme chybějící soubory například pomocí `svn add` přidat.

```
svn status
```

Změny odešlete je pomocí `commit` do repozitory.

```
svn commit -m "Text zprávy"
```

Postup se opakuje, jen `update` nemusíte dávat pokaždé pokud jste si jistý, že nikdo s repozitorem mezitím nepracoval. Případně můžete zamknout repozitorem během práce na změnách. Pokud nenastane konflikt vystačíte si s tímto jednoduchým postupem. Pokud dojde ke konfliktu můžete změny spojit (`merge`) nebo svoje vrátit pomocí `svn revert`.

5.4. Export pro hosting, ftp apod.

Pokud by vám vadí adresáře, které `svn` vytváří automaticky a uchovává v nich svoje informace, tak můžete se zbavit pomocí exportu.

Pokud chcete do aktuálního adresáře všechny soubory vyexportovat stačí zadat tento příkaz a soubory se tam uloží. Dá se také pracovat s pracovní kopii nebo zadávat i cestu pro export.

```
svn export file:///C:/rep/rep_test/trunk
```

Příklad exportního skriptu v shellu linux pro export projektu. Exportuje se poslední revize, a vytvoří se kromě exportu i balík označený revizí a datumem vytvoření.

```
# path to export files
EXPORTPATH=/home/user/export/projekt
# url of repository
REPOS=file:///home/user/repos/projekt/trunk/
# path to repository
REPOSPATH=/home/user/repos/projekt/
```

```
# last revision
REV="$( svnlook youngest $REPOSPATH )"
# make export path
mkdir -p $EXPORTPATH
# svn force export (nothing to screen)
svn export --force $REPOS $EXPORTPATH >> /dev/null
cd $EXPORTPATH
cd ..
# make tar.gz file for projekt
tar -czvf projekt-R$REV-`date +%Y%m%d-%H%M%S`.tgz projekt >> /dev/null
```

Pokud budete chtít vyexportovat jen soubory co se změnili v poslední revizi bude to trochu složitější a budete potřebovat tento skript.

```
# path to export files
EXPORTPATH=/tmp/test/
# url of repository
REPOS=file:///home/user/repos/projekt/
# path to repository
REPOSPATH=/home/user/repos/projekt/
# last revision
REV="$( svnlook youngest $REPOSPATH )"

# function to list and export file by file
pathexport() # $1
{
  mkdir -p $EXPORTPATH${2%/*}
  svn export --force $1$2 $EXPORTPATH$2
}

# make export path
mkdir -p $EXPORTPATH
# list all changed files
for i in $( svnlook changed -r $REV $REPOSPATH ); do
  if [ "${#i}" -gt "2" ]
  then pathexport $REPOS $i
  fi
done
```

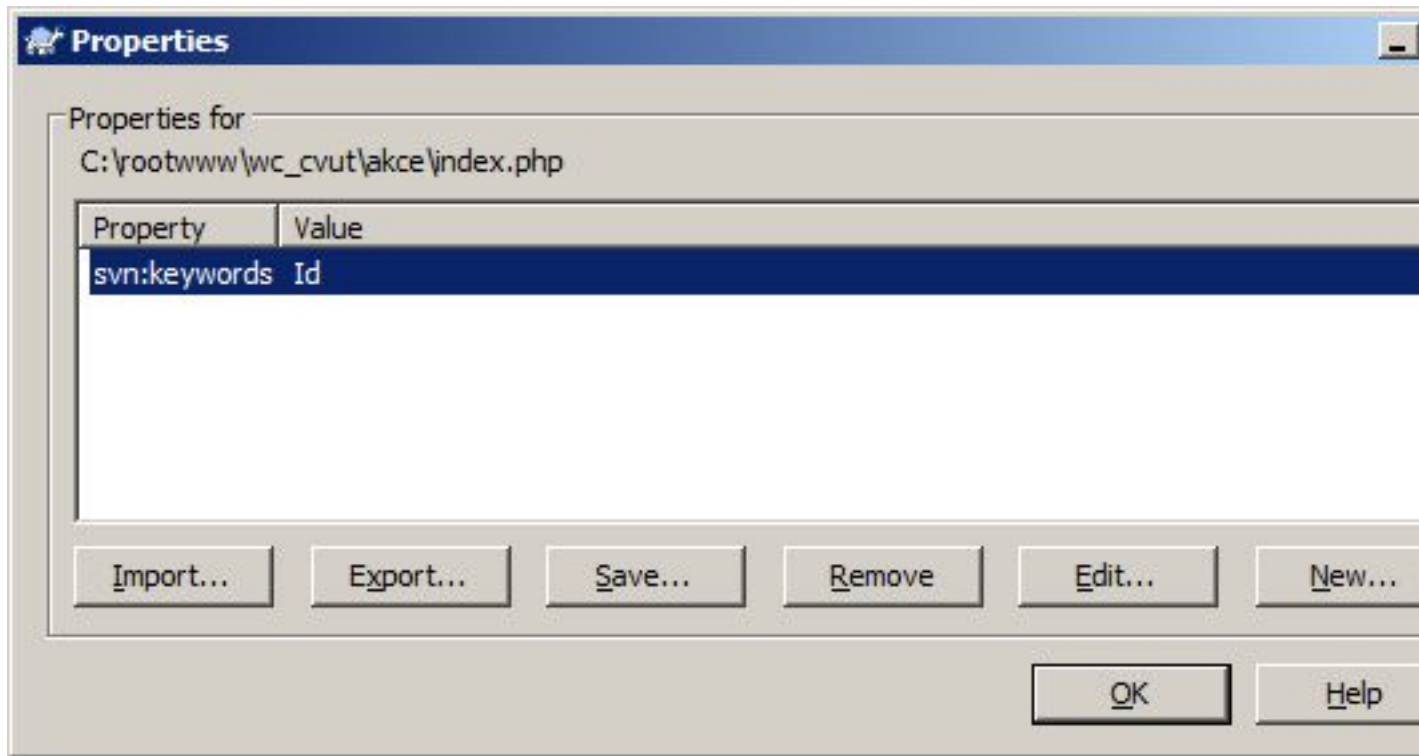
6. Pokročilejší funkce

6.1. Vlastnosti

Vlastnosti (properties) umožňují uchovávat další informace o jednotlivých souborech. Nastavují se automaticky nebo ručně.

Mezi ty automatické patří od verze 1.5 takzvaný "merge tracking". Některí klienti ho umí používat a zobrazují potom informace o tom kde co bylo spojeno.

Určit si každý kdo listoval v projektech, které jsou zpracovávány pod Subversion, v hlavičce informace o verzi a autorovy (\$Id: svn-kniha.xml 50 2009-07-20 15:25:15Z abtris \$). Tato informace je generována automaticky pomocí vlastnosti svn:keywords=Id. Kromě Id může tato vlastnost nabývat více možností a to Author, Date, Revision a HeadURL. Pro verzi dokumentu se ale obvykle používá Id.

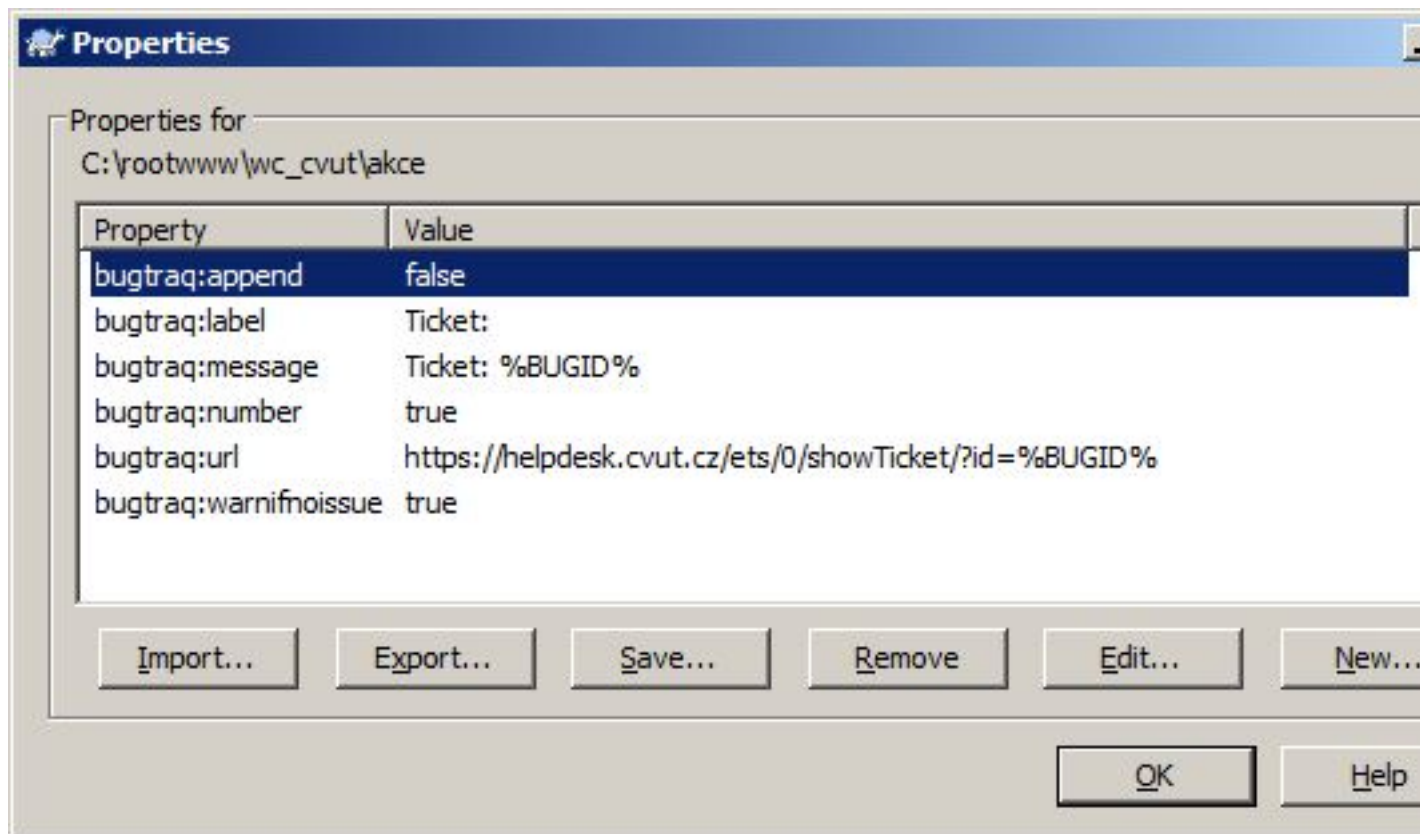


Příklad použití v PHP

```
<?php
/**
 * Pridani obecne
 * @version SVN: $Id: svn-kniha.xml 50 2009-07-20 15:25:15Z abtris $
 * @package Akce2008
 * @filesource
 */
```

6.2. Vlastnosti projektu

U projekt# se dá nastavit například integrace s systémem pro správu chyb (bug, issue tracker) a může být zcela libovolný, klidně jednoduchá aplikace napsaná v PHP nebo i profesionální jako je JIRA [<http://www.atlassian.com/software/jira/>].



6.3. Migrace repozitory

Celé repozitory můžete uložit do souboru.

```
svnadmin dump repository >file
```

Celý soubor jde stejně dobře nahrát.

```
svnadmin load repository < file
```

Ve windows můžete udělat export všech repozitory například pomocí tohoto skriptu (přepínač --incremental revize jako diff proti předchozí verzi).

```
dir c:\rep /b /O /AD >dir1.txt
```

```
FOR /F %%M IN (dir1.txt) DO ( ECHO %%M svnadmin dump c:/rep/%%M --incremental >%M.dmp zip -m -9 -g %%M.dmp.zip %%M.dmp )
```

6.4. Mirror repository pomocí svnsync

Pokud potřebujete read-only mirror, například veřejné dostupnosti v případě Open Source projektů nebo jako zálohu má SVN od verze 1.4 nástroj svnsync.

V praxi budete nejdříve mirror inicializovat a potom pravidelně spouštět synchronizaci, nejlépe post-commitem nebo cronem.

Pokud děláte mirror z nějakého staršího svn repozitory (existuje již několik let) tak můžete narazit na několik problémů s kterými poradím co dělat.

Inicializace mirroru

```
svnsync init file:///srv/svn/rep-mirror https://svn.firma.cz/svn/rep-zdroj \
--source-username svnuser --source-password svnpass \
--sync-username svnmirroruser --sync-password svnmirrorpass
```

Vlastní synchronizace

```
svnsync synchronize file:///srv/svn/rep-mirror \
--source-username svnuser --source-password svnpass \
--sync-username svnmirroruser --sync-password svnmirrorpass
```

6.4.1. #ešení možných problém#

Příkazy je lepší psát na jeden řádek, ale v linuxu by nemělo vadit ani rozdělení na více. U synchronizace mi to vyhodilo chybu a byl jsem nucen to dát na jeden řádek. Tady to pro přehlednost nechávám.

```
svnsync: Cannot accept 'svn:log' property because it is not encoded in UTF-8
```

Tato chyba bývá způsobena špatnými znaky v logu například češtinou z windows kódování a je potřeba opravit záznam v svn:log.

Oprava log message se provede buď pomocí svn propset -r111 --revprop svn:log -F fixedlogfile nebo svnadmin setlog /srv/svn/rep-zdroj -r 111 fixedlogfile. Pro svn propset musíte mít povolený a nastavený pre-revprop-change hook skript.

```
svnsync: Cannot accept non-LF line endings in 'svn:log' property
```

Pokud se vyskytne tato chyba je potřeba upgradovat SVN server na verzi 1.6.3, kde už umí převádět konce řádků automaticky.

6.5. Branches a tags - větve a štítky

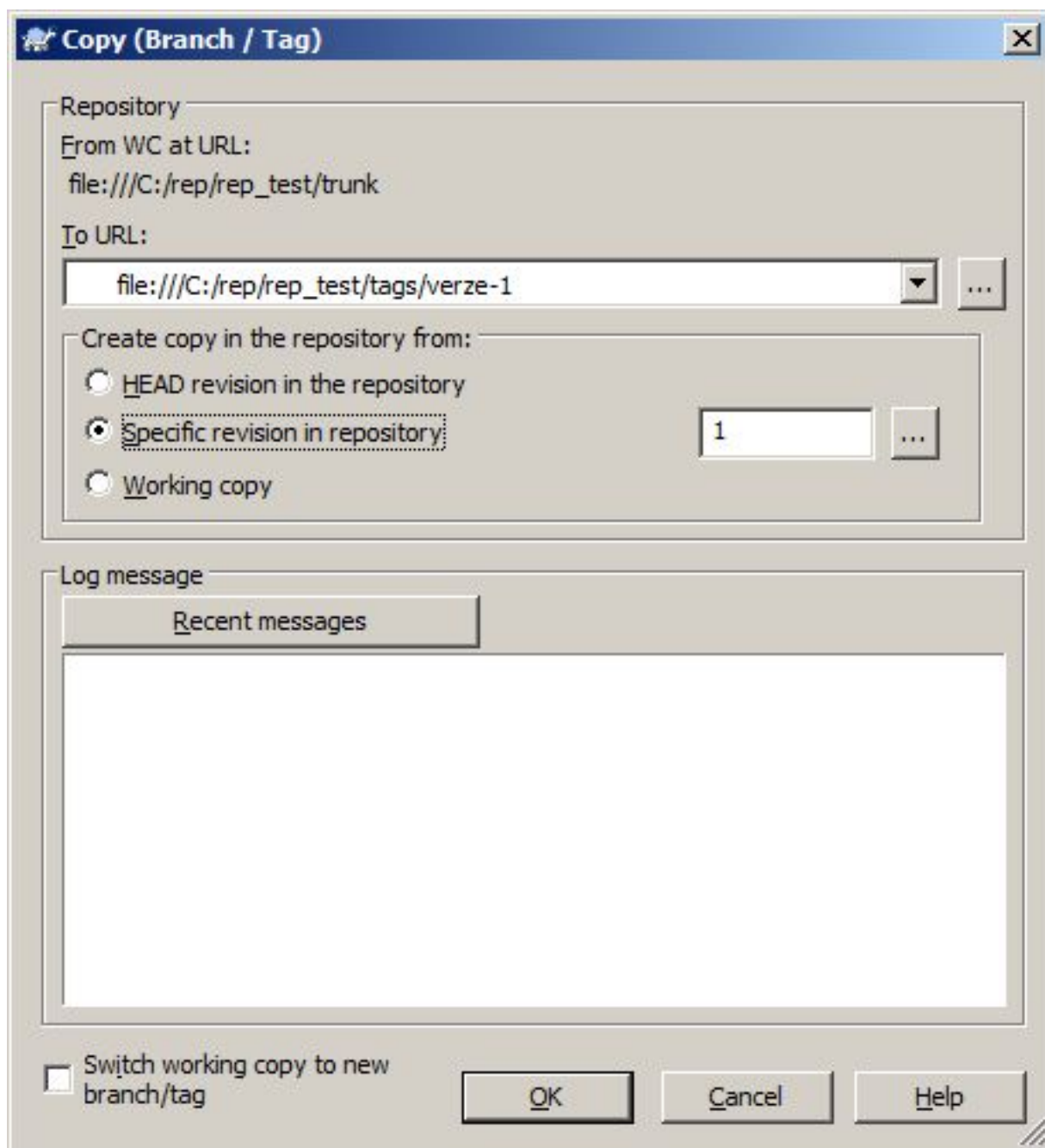
Tagy slouží pro zachování určité verze repozitory v #ase. V angličtině tomu říkájí snapshot. Udělá se jednoduše kopie vaší verze kterou chcete označit štítkem.

```
svn copy file:///C:/rep/rep_test/trunk file:///C:/rep/rep_test/tags/verze-1 \
-m "Verze 1 mého zkušebního projektu."
```

Tagy nemusí představovat nutně jen kopii v repozitory, ale můžete zachovat i například současnou pracovní kopii (wc). To se hodí pokud je projekt obsáhlý nebo projevuje chyby na které te# nemáte čas, ale chcete se k nim někdy vrátit.

```
svn copy c:/wc/test file:///C:/rep/rep_test/tags/problematicka-wc \
-m "Kopie mojí wc."
```

V TSVN je situace jednoduchá tam máte příkaz Branch/tag pomocí kterého vytvoříte příslušný tag nebo branch (větve). Jak je vidět z následujícího obrázku máte k dispozici několik voleb, které vám dávají možnost vytvořit kopii z HEAD (poslední verze v repozitory) nebo s libovolné verze kterou specifikujete číslem revize a nakonec z příslušné pracovní kopie projektu. Dole pak je volba, která umožní vám nejenom udělat kopii, ale také přepnout WC na vámi udělaný branch/tag. To se používá pokud vytváříte branch a chcete v ní pokračovat.



Pokud už víte co jsou tagy také vás nepřekvapí, že branches jsou v rámci repozitory jsou to samé jen se s nimi trochu jinak pracuje. Tagy slouží k zaznamenání určité verze, ale branch slouží k tomu aby se dále upravovala, opravovala apod.

Uvedu jeden příklad například projekt má 2 verze. Na verzi 2.0 se pracuje v /trunk a verze 1.0 je vydána v /tags/v1-0. Teď ale někdo najde ve verzi 1.0 chybu tak se vytvoří branch v /branches/1-0/ a pracuje se na odstranění chyby, udělá se několik commitů, ty se otestují a když je vše v pořádku vydá se verze v tags /tags/v1-1. Nezávisle na tom se pracuje na verzi 2. Možných schémat práce s verzemi je několik, toto byl jen příklad.

6.6. Merge v příkladech

Příkaz merge slouží ke spojování zdrojových kódů, v praxi většinou nastávají tyto případy:

1. Merge zm#n z n#jaké branche do trunku. Pokud jste to opravili na jednom míst# takto to p#enesete i jina#m.

```
svn merge --reintegrate file:///C:/rep/rep_test/branches/test
```

2. Reverzní merge, návrat ve trunku nebo v#tvi k p#edchozí verzi. Vrá#íte zp#t zm#ny, které jste ud#lali mezi revizemi HEAD:80.

```
svn merge -r HEAD:80 file:///C:/rep/rep_test/trunk
```

3. Kompletní merge do pracovní kopie. Pomocí syntaxe URL@REV [mailto:URL@REV] se dá p#ímo zadat revize. První URL (levá strana pro porovnání) a druhé URL (pravá strana pro porovnání) a cíl (pracovní kopie).

```
svn merge file:///C:/rep/rep_test/branches/test@150 \  
file:///C:/rep/rep_test/branches/jiny-test@212 \  
c:/wc/test
```

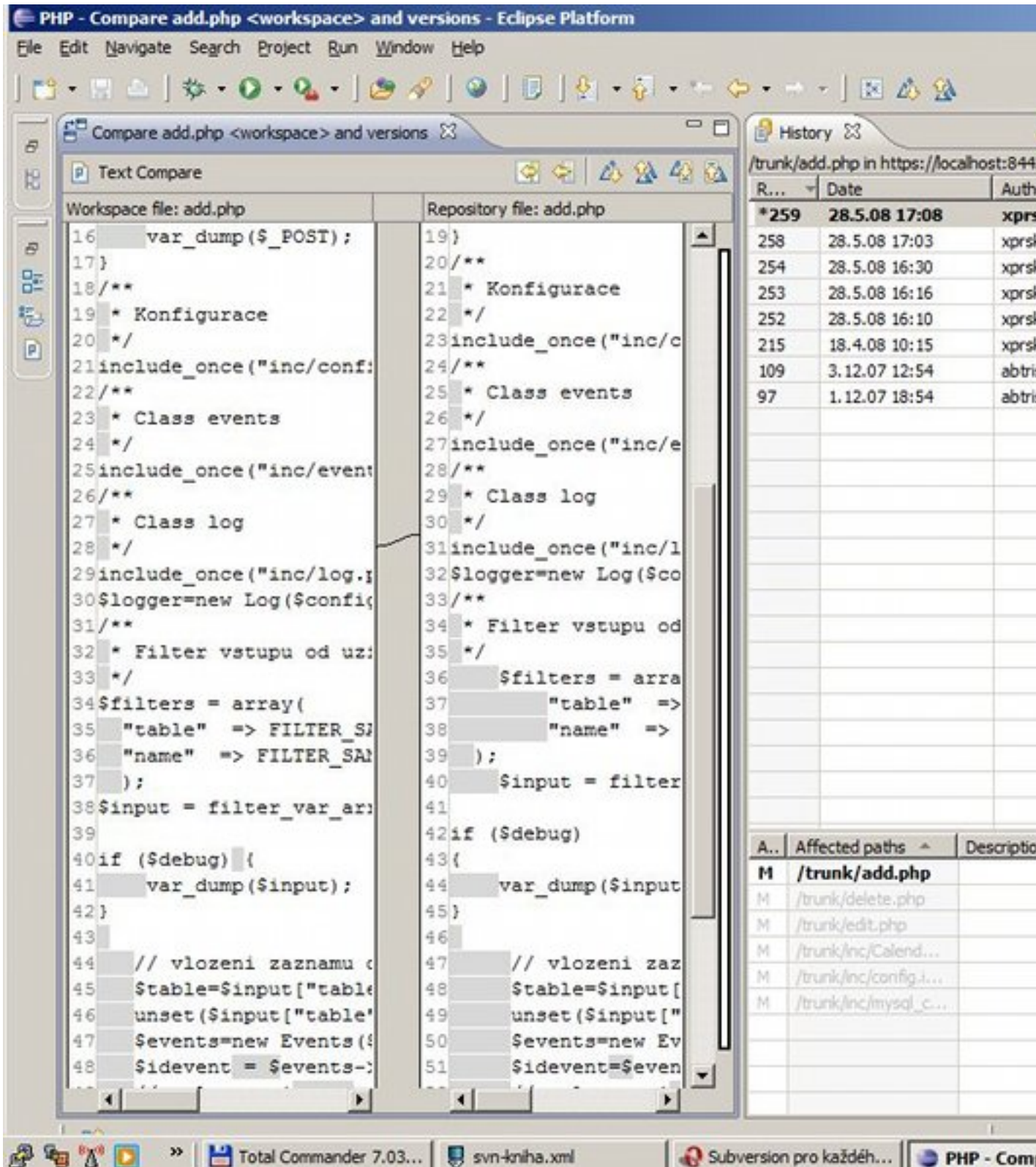
Pokud provádíte merge z p#íkazové #ádky, hodí se také p#epína# --dry-run s kterým si merge vyzkoušíte nane#isto. Informace o merge se dá zjistit také pomocí p#íkazu `svn mergeinfo`.

7. Integrace SVN s editory a IDE

7.1. Eclipse

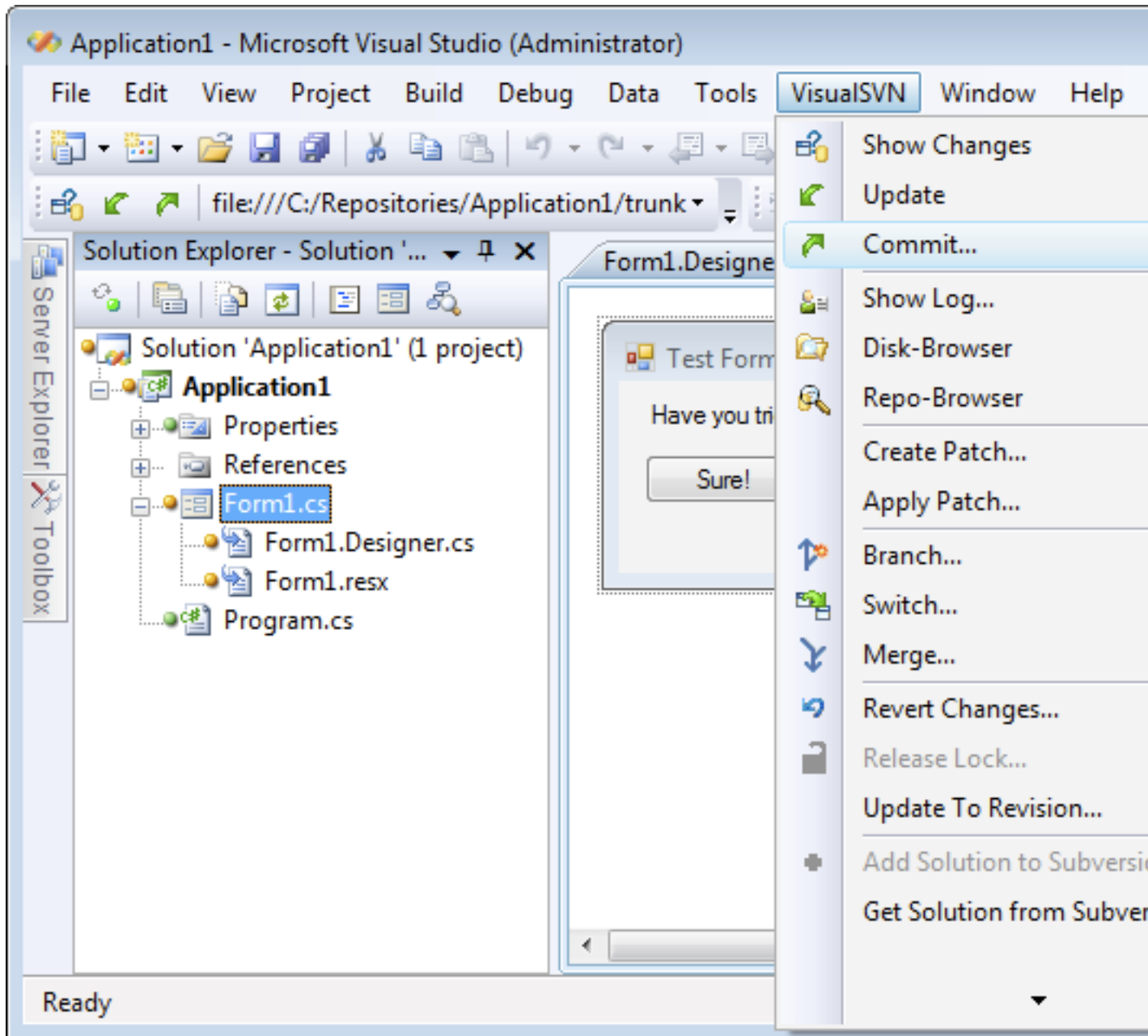
Eclipse má nativn# podporu jen pro CVS, pro Subversion musíte nainstalovat plugin Subclipse [<http://subclipse.tigris.org>] nebo Subversive [<http://www.eclipse.org/subversive/>], který používá nap#íklad Zend Studio for Eclipse

Po delší zkušenosti s Zend Studi#m doporu#uji všem spíše Subversive.



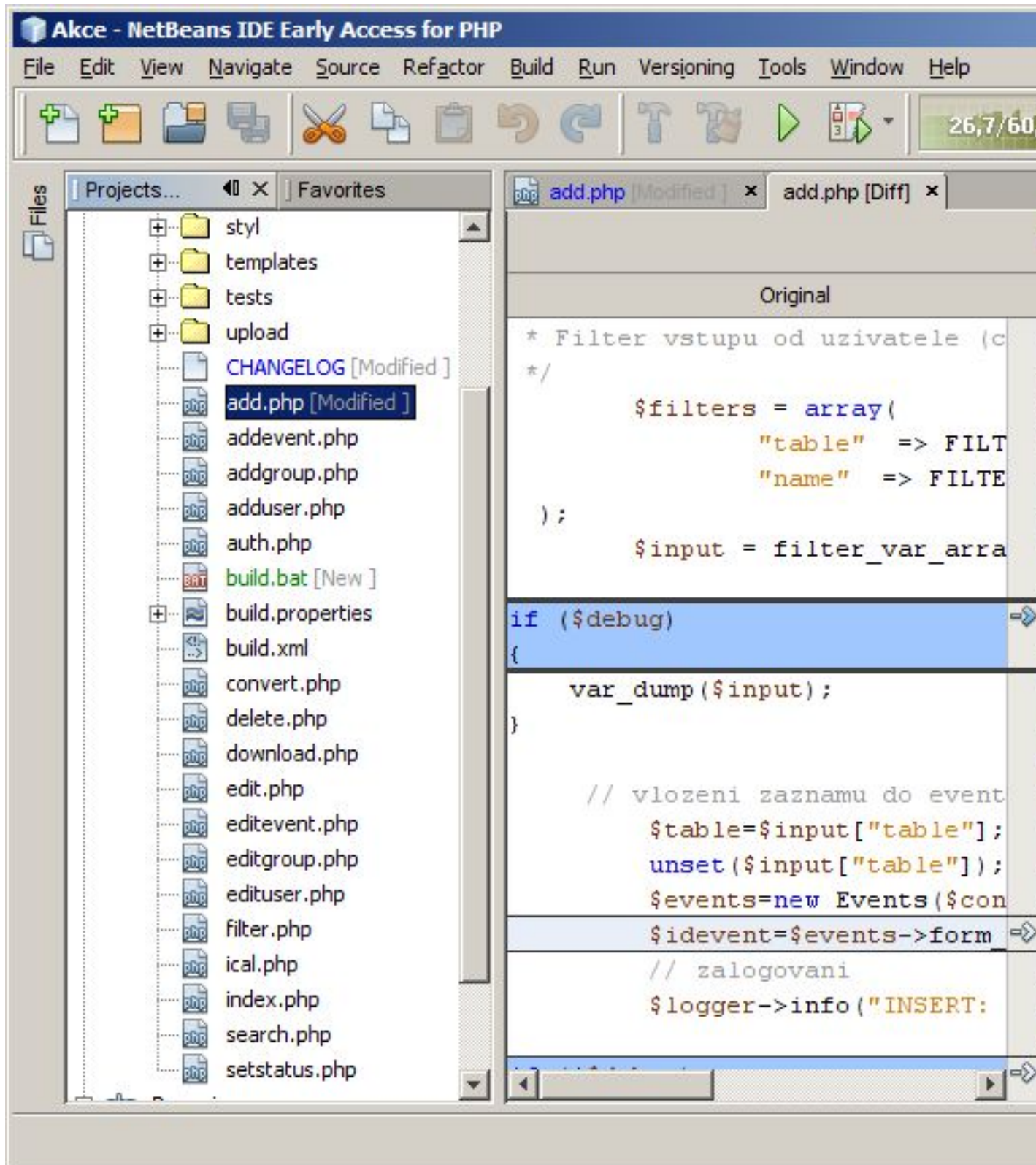
7.2. Microsoft Visual Studio

Pro práci se Subversion budete potřebovat AnkhSVN 2.0 [<http://ankhsvn.open.collab.net/>] nebo VisualSVN [<http://www.visualsvn.com/visualsvn/>]

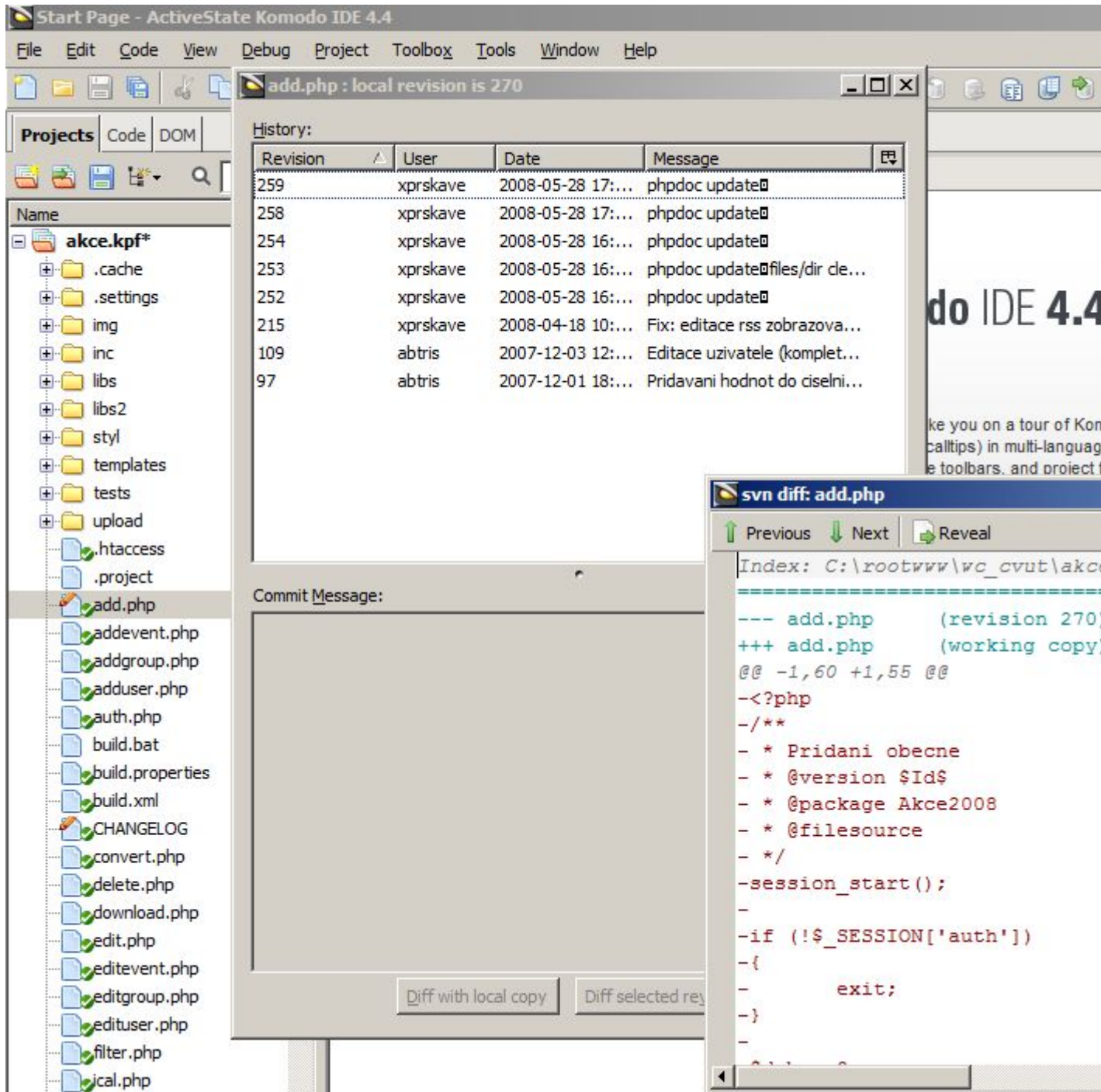


7.3. NetBeans, Komodo IDE, SciTE a další

SVN #ádkový klient se používá v mnoha editorech i v IDE. Jako ukázkou tu mám obrázek z NetBeans [<http://www.netbeans.org/>], které poskytují celkem p#kný komfort a využívají jen #ádkového klienta což p#inese výhodu nap#. p#i vydání nové verze Subversion, sta#í aktualizovat SVN klienta a nemusíte shán#t podporu pro svoje IDE.



Komodo IDE [http://www.activestate.com/Products/komodo_ide/] patří ke komerčním editorům, který podporuje mnoho jazyků (PHP, Perl, Python, Ruby, Tcl, Javascript, Actionscript) a ze SCM podporuje kromě Subversion, Perforce a CVS.



7.4. JetBrains (RubyMine, PHPStorm, IntelliJ IDEA)

JetBrains [<http://www.jetbrains.com/products.html>] mají velmi kvalitní IDE pro Javu, PHP, Ruby a další, kde je vestavěná podpora pro SVN, Git a další VCS.

8. Distribuované systémy pro správu verzí

8.1. Přechod od Subversion k Mercurial

Mercurial (hg) je další distribuovaný systém, který je napsaný v pythonu a podporuje ho například google ve svém code.google.com [<http://code.google.com>] a také existuje hosting bitbucket.org [<http://bitbucket.org>], který umožňuje zdarma hosting open source projektů jako code.google.com [<http://code.google.com>] nebo github.com [<http://github.com>] pro git. Služby github.com a bitbucket.org mají také své placené varianty pro komerční využití.

Keywords v hg normálně nejsou, řeší se to přes standardní doplněk, který je potřeba zaplnout a nastavit v konfiguračním souboru .hgrc.

```
[extensions]
hgext.keyword =
keyword=/usr/share/python-support/mercurial-common/hgext/keyword.py
[keyword]
**.php =
**.xml =
[keywordmaps]
RCSFile = {file|basename},v
Author = {author|user}
Header = {root}/{file},v {node|short} {date|utcdate} {author|user}
Source = {root}/{file},v
Date = {date|utcdate}
Id = {file|basename},v {node|short} {date|utcdate} {author|user}
Revision = {node|short}
```

Export - obdoba svn export je v hg příkaz archive, který ale umožňuje také vytvořit ze souborů přímo archív (zip, tar, tgz, tbz2).

```
hg archive path/to/dir/
hg archive -t zip /path/to/file.zip
```

Log v xml formátu kompatibilním se svn log --xml. Pro logování jsem vytvořil script, který umí to co standardní parametr xml v SVN. Parametrem scriptu je výstupní soubor.

```
# http://hgbook.red-bean.com/read/customizing-the-output-of-mercurial.html
echo '<?xml version="1.0"?>\n<log>\n' >$1.xml
hg log --template '<logentry revision="{rev}">
<author>{author|obfuscate}</author>
<date>{date|isodate}</date>
<msg>{desc|escape}\n</msg>
<paths><path>{files}</path></paths>
</logentry>\n' $1/ >>$1.xml
echo '</log>\n' >>$1.xml
```

8.2. Spolupráce distribuovaných systému se Subversion

Některé distribuované systémy (git, bazaar a jistě další) mají utility, které umožňují, že lokálně pracujete jako s distribuovaným systémem, ale push provádíte do SVN repozitory.

Ukázka postupu v gitu (git-svn)

```
git svn clone file:///srv/svn/repos/test
...
n#jaké zm#ny
...
git commit -a
git svn dcommit
```

Ukázka postupu v bazaar (bZR)

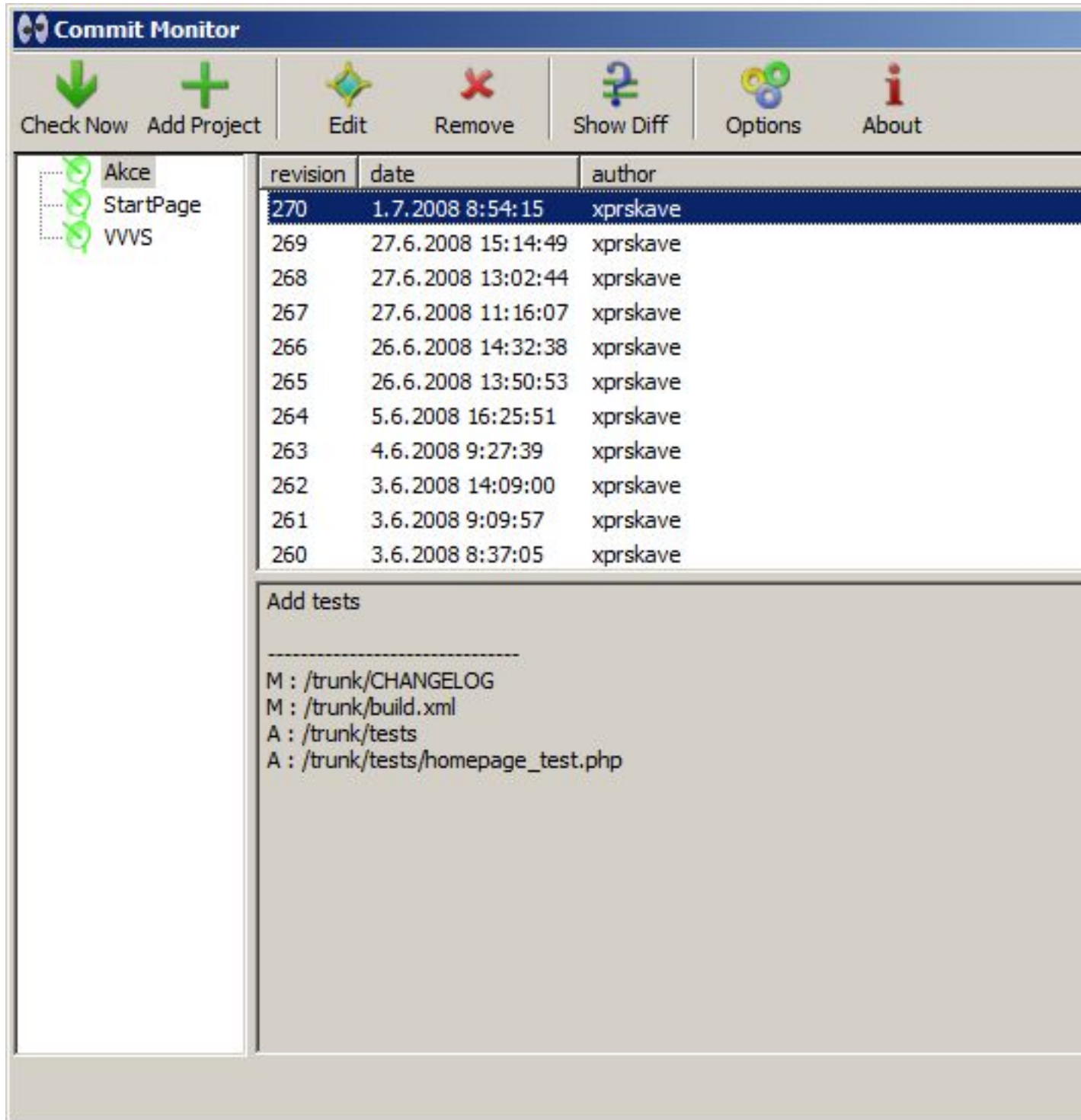
```
bZR checkout file:///srv/svn/repos/test
bZR unbind
...
n#jaké zm#ny
...
bZR commit
bZR bind
bZR push file:///home/svn/repos/test
```

Tento postup lze zkrátit pomocí `bZR commit --local` potom nemusíme použít `bind` a `unbind`.

9. Nástroje pro práci se Subversion

9.1. Commit Monitor

Program Commit Monitor pro Windows umožňuje sledovat změny v repozitory, které si do něj zadáte a upozorní vás pokud někdo provede commit. Autorem je Stefan Küng (<http://tools.tortoisesvn.net/>). Podobný program je ještě například SVN Notifier.



Bohužel neznám podobnou aplikaci pro Linux, obvykle se to nahrazuje mailovou konferencí kam chodí automaticky všechny commity pro projekt.

Existují webové nástroje, které podobné informace poskytují o projektech. Google Code [<http://code.google.com/>] a Github [<http://github.com>] například poskytuje RSS o změnách. Podobně třeba Trac [<http://trac.edgewall.org/>] nebo jiné systémy, které mají podporu pro SVN.

Případně si to můžete udělat sami, jako například já v PHP [<http://code.google.com/p/php-svn-changelog/>].

9.2. StatSVN

StatSVN je program napsaný v Jav# a slouží k analýze log#. Stáhnout si ho můžete na <http://sourceforge.net/projects/statsvn/>.

Použití je potom jednoduché jako první parametr dáte log v xml formátu a jako druhý cestu k aktuální pracovní kopii a potom výstupní adresá# pro generované html soubory.

```
java -jar statsvn.jar c:\rep\test\test_changelog.xml \
c:\rep\test\ -output-dir c:\tmp\test
```

Výstup potom vypadá jako na obrázku. Tento program je zdarma, ale umí v#tšinu statických v#cí podobn# jako profesionální programy pro práci s repozitory jako je FishEYE [<http://www.atlassian.com/software/fisheye/>] i když není zdaleka tak v líbivém kabát#.

Development Statistics for /trunk

Generated: 2008-06-18 15:00

Report Period: 2007-11-29 to 2008-06-05

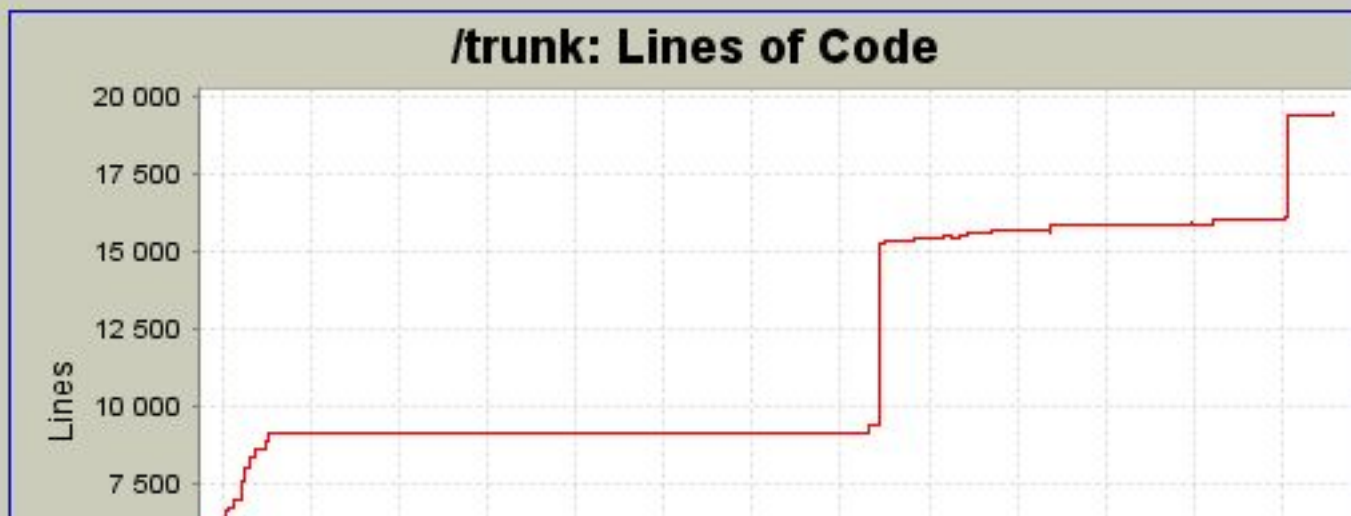
Total Files: 105

Total Lines of Code: 19436

Developers: 2

- [Developers](#)
- [Commit Logs](#)
- [Lines of Code](#)
- [File Statistics](#)
- [Directory Sizes](#)
- [Repo Heatmap](#)
- [LOC and Churn](#)

Lines of Code



9.3. Generování changelogu pomocí svn2cl

Subversion utilita `svn2cl` [<http://arthurdejong.org/svn2cl/>] je b#žn# dostupná v balíku `subversion-tools` v Debianu/Ubuntu.

Nap#íklad pro generování changelogu pro tuto knihu se d#lal takto:

```
/usr/bin/svn2cl --strip-prefix=trunk \  
  --break-before-msg=2 \  
  --group-by-day \  
  --separate-daylogs \  
  -i \  
  -o changelog.txt
```

```
--authors=authors.xml \  
file://localhost/svn/svn-kniha/trunk/ \  
-o changelog.html \  
--html
```

Soubor authors.xml obsahuje toto:

```
<?xml version="1.0" encoding="utf-8"?>  
  <authors>  
    <author uid="abtris"> Ladislav Prskavec &lt;ladislav@prskavec.net&gt;  
  </author>  
</authors>
```

Výsledek v textové podobě vypadá nějak takto:

```
2009-06-17  Ladislav Prskavec <ladislav@prskavec.net>  
  
* [r35] changelog, changelog.html, html/apa.html, html/bi01.html,  
html/ch01.html, html/ch01s02.html, html/ch02.html,  
html/ch02s02.html, html/ch02s03.html, html/ch02s04.html,  
html/ch02s05.html, html/ch03.html, html/ch03s02.html,  
html/ch03s03.html, html/ch04.html, html/ch04s02.html,  
html/ch04s03.html, html/ch04s04.html, html/ch05.html,  
html/ch05s02.html, html/ch05s03.html, html/ch05s04.html,  
html/ch05s05.html, html/ch06.html, html/ch06s02.html,  
html/ch06s03.html, html/ch07.html, html/ch07s02.html,  
html/index.html, html/pr01.html, index.html, make.sh,  
svn-kniha.html, svn-kniha.xml, svn2html.css:  
  
Update generating.  
Link to changelog.  
  
* [r34] authors.xml, changelog, html/apa.html, html/bi01.html,  
html/ch01.html, html/ch01s02.html, html/ch02.html,  
html/ch02s02.html, html/ch02s03.html, html/ch02s04.html,  
html/ch02s05.html, html/ch03.html, html/ch03s02.html,  
html/ch03s03.html, html/ch04.html, html/ch04s02.html,  
html/ch04s03.html, html/ch04s04.html, html/ch05.html,  
html/ch05s02.html, html/ch05s03.html, html/ch05s04.html,  
html/ch05s05.html, html/ch06.html, html/ch06s02.html,  
html/ch06s03.html, html/ch07.html, html/ch07s02.html,  
html/index.html, html/pr01.html, make.sh, svn-kniha.html:  
  
Update generating, add make changelog  
  
...
```

10. Literatura a použité zdroje

[SVN] Subversion. URL: <http://subversion.tigris.org/>.

[BZR] Bazaar VCS. URL: <http://bazaar-vcs.org/>.

[WIKISVN] Wikipedia Subversion. URL: <http://cs.wikipedia.org/wiki/Subversion/>.

[DB] Docbook. URL: <http://www.docbook.cz>.

10.1. SVN klienti

[TSVN] TortoiseSVN. URL: <http://tortoisesvn.tigris.org/>.

[SUBCLIPSE] Subclipse. URL: <http://subclipse.tigris.org/>.

[SUBVERSIVE] Subversive. URL: <http://www.eclipse.org/subversive/>.

[ANKHSVN] AnkhSVN. URL: <http://ankhsvn.open.collab.net/>.

[RSVN] RapidSVN. URL: <http://rapidsvn.tigris.org/>.

[VISUALSVN] VisualSVN : Subversion for Visual Studio. URL: <http://www.visualsvn.com/visualsvn/>.

10.2. SVN servery

[VSVN] VisualSVN server. URL: <http://www.visualsvn.com/server/>. Programovací prostředí - IDE

[ECLIPSE] Eclipse. URL: <http://www.eclipse.org/>.

[ZEND] Zend Studio for Eclipse. URL: <http://www.zend.com/en/products/studio/>.

[NB] NetBeans. URL: <http://www.netbeans.org/>.

[KOMODO] Komodo IDE. URL: http://www.activestate.com/Products/komodo_ide/.

[VS] MS Visual Studio. URL: <http://msdn.microsoft.com/cs-cz/vstudio/>.

10.3. Dokumentace

[SVNBOOK] SVN book. URL: <http://svnbook.red-bean.com/>.

11. Používané zkratky a anglické příkazy

DIR - adresář

FILENAME - název souboru

SVN - Subversion

TSVN - TortoiseSVN

REV - revize

ARG - argumenty

NAME - název

VALUE - hodnota

MESSAGE - zpráva

NUM - pořadí

PATH - cesta k souboru, repozitory, pracovní kopii

WC - pracovní kopie (working copy)

Alias - zkratka