# SubVersion (SVN)


Didier DONSEZ

Université Joseph Fourier - Grenoble 1

PolyTech'Grenoble LIG/ADELE

**x.y@imag.fr, x.y@ieee.org, y@apache.org**

**with x=didier,y=donsez**

# Licence

- Cette présentation est couverte
  par le contrat Creative Commons By NC ND
  - http://creativecommons.org/licenses/by-nc-nd/2.0/fr/

# Motivations

- SCM
  - Gestion de versions/revisions de projets logiciels
    - Collaboratif, Distribuée
- Limitations de CVS
  - Pas de versionnement des répertoires
  - Pas de notion de destruction, de déplacement, de copie de sous-arborescences
    - Par exemple : l'historique transmis aux éléments supprimés puis recréés
  - Révision par fichier
  - Commit non atomique (révision groupée de plusieurs fichiers)
  - Manque de métadonnées versionnables
  - Stockage intégrale des fichiers binaires (pas de *diff* binaire)
  - Manque d'un protocole orienté Web

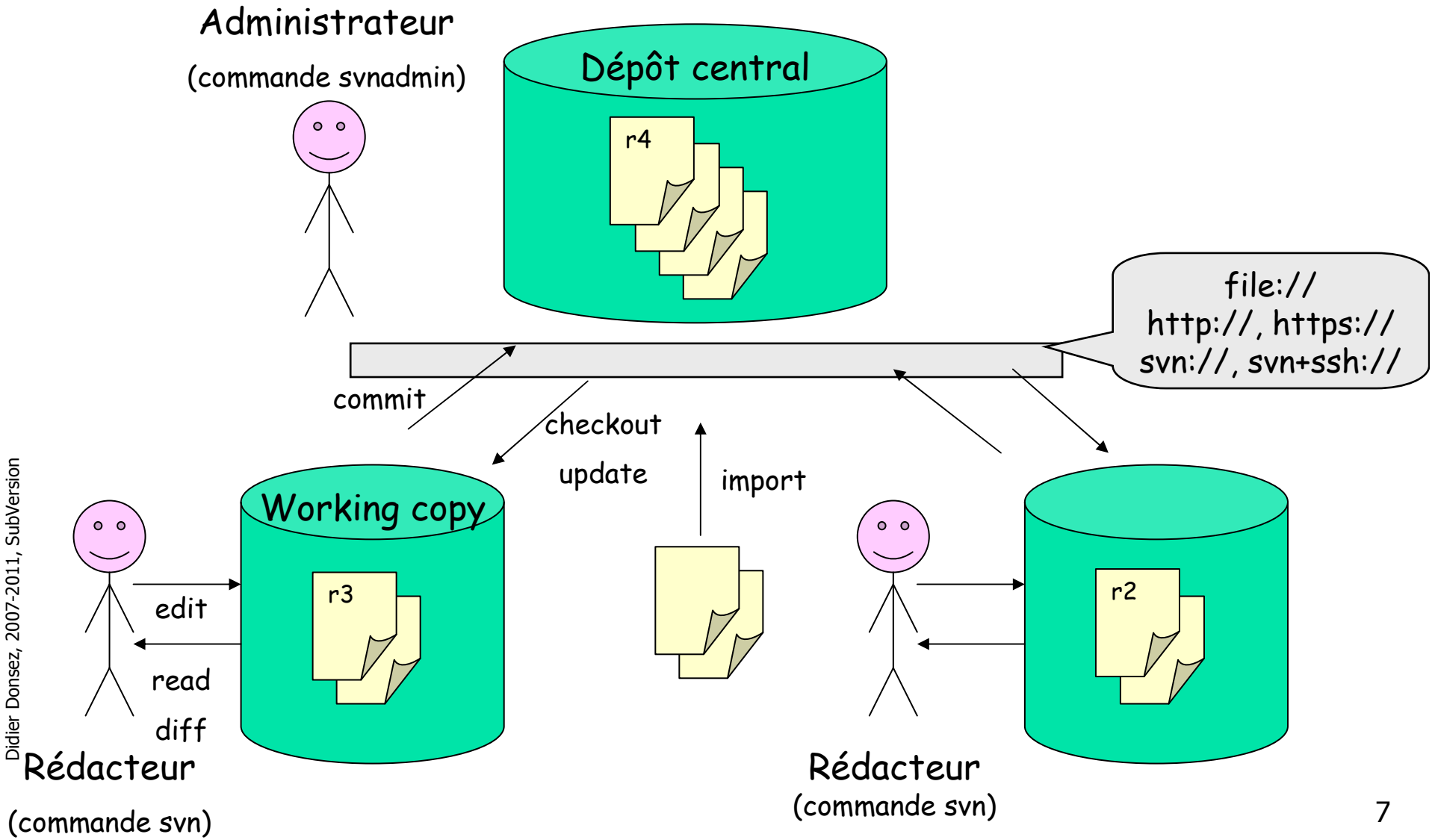
# Principales caractéristiques (fonctionnelles) de SubVersion

- Concurrence optimiste (Copy-Modify-Merge)
  - Mais possibilité de verrouillage
- Révision par groupe de modifications
- Révision de fichiers, de répertoires et de métadonnées
- Commit/Import/Update atomique
  - révision groupée de plusieurs fichiers
- Notion de destruction et de déplacement, de copie de sous-arborescences
- Propriétés (métadonnées sur fichiers et répertoires)
  - révisables, spéciale
- Pas de notion de tags ou de branches

# Principales caractéristiques (techniques) de SubVersion

- Stockage différentiel des fichiers binaires
- Copies faignantes des arborescences
- Fonctionnement offline
  - car gros espaces sur les stations et utilisation en contexte Web
  - Répertoires `./.svn`
- Echange différentiel entre client et serveur
- Plusieurs modes d'accès
  - Systèmes de fichiers (`file:`)
  - WebDAV/DeltaV (`http:,https:`)
  - protocole propriétaire (`svn:,svn+ssh:`)
- Navigation et utilisation (*commit*) via des clients WebDAV et WebDAV/DeltaV
  - Support de l'auto-versionning pour les clients WebDAV

# Architecture de SubVersion



# Principales commandes

- Rédacteur
  - svn
  - svnlook
  - svnshell
- Administrateur
  - svnadmin
  - svndumpfilter
  - svnlook
  - svnshell
- Serveurs
  - svnserve, mod_dav_svn pour Apache HTTPD

# Révisions

- Révision par groupe d'actions
  - modification/création/suppression/déplacement de fichiers/répertoires/métadonnées
  - différent de CVS
    - Une par fichier
  - Numéro entier croissant
    - rev=0 à la création du dépôt
      - Une nouvelle révision créée à chaque nouveau commit
- Raccourcis

  HEAD → The latest (or "youngest") revision in the repository.

  BASE → The revision number of an item in a working copy.

  COMMITTED → most recent revision prior to, or equal to, BASE, in which an item changed.

  PREV → immediately *before* the last revision in which an item changed. = COMMITTED-1.

# Exemples d'utilisation des raccourcis

- svn diff -r PREV:COMMITTED Hello.java
  - shows the last change committed to Hello.java
- svn log -r HEAD
  - shows log message for the latest repository commit
- svn diff -r HEAD
  - compares your working copy (with all of its local changes) to the latest version of that tree in the repository
- svn diff -r BASE:HEAD Hello.java
  - compares the unmodified version of Hello.java with the latest version of Hello.java in the repository
- svn log -r BASE:HEAD
  - shows all commit logs for the current versioned directory since you last updated
- svn update -r PREV Hello.java
  - rewinds the last change on Hello.java, decreasing Hello.java's working revision
- svn diff -r BASE:32 Hello.java
  - compares the unmodified version of Hello.java with the way Hello.java looked in revision 32

# Exemples d'utilisation des dates pour les révisions

- svn checkout -r {2007-09-17}
- svn checkout -r {16:30}
- svn checkout -r {16:30:00.200000}
- svn checkout -r {"2007-09-17 16:30"}
- svn checkout -r {"2007-09-17 16:30 +0230"}
- svn checkout -r {2007-09-17T16:30}
- svn checkout -r {2007-09-17T16:30Z}
- svn checkout -r {2007-09-17T16:30-04:00}
- svn checkout -r {20070917T1630}
- svn checkout -r {20070917T1630Z}
- svn checkout -r {20070917T1630-0500}

- Remarque
  - "2007-09-17" == "2007-09-17 00:00"
    - Référence seulement les révisions faites avant le 16 à minuit

# Création d'un dépôt

- Création
  - svnadmin create c:\repository
    - l'option --fs-type permet de spécifier le système de fichiers utilisés pour le stockage (BerkeleyDB par défaut ou FSFS)
- Importation initial
  - des fichier du projet hello dans le dépôt
  - svn import hello file:///c:/repository/hello --message "initial import"
- Inspection
  - svnlook youngest repository
  - svnlook tree repository (~ --revision HEAD)
  - svn info file:///c:/repository/hello
  - svn log –r10 file:///c:/repository/hello
  - svn cat –r10 file:///c:/repository/hello/README.txt

# Création d'une copie de travail

- Création (checkout)
  - svn checkout file:///c:/repository/hello workingcopy1
  - svn status workingcopy1 --verbose
- Ajout, renommage et suppression de fichiers
  - echo This is a readme file > workingcopy1\LISEZMOI.txt
  - svn add workingcopy1\LISEZMOI.txt
  - svn rename workingcopy1\LISEZMOI.txt workingcopy1\README.txt
  - svn delete workingcopy1\LICENCE.txt
  - svn status workingcopy1 --verbose
- Validation des modifications auprès du dépôt
  - svn commit workingcopy1 --message "some modifications"
- Synchronisation avec la dernière révision du dépôt
  - svn update workingcopy1

# Status des fichiers

- A → Resource is scheduled for Addition
- D → Resource is scheduled for Deletion
- M → Resource has local Modifications
- C → Resource has Conflicts
  - changes have not been completely merged between the repository and working copy version
- X → Resource is eXternal to this working copy
  - may come from another repository.
- ? → Resource is not under version control
- ! → Resource is missing or incomplete
  - removed by another tool than SubVersion

# Edition collaborative

- Conflit lors d'un commit (status=C)
  - Un autre rédacteur a modifié et validé un fichier qui est en cours de validation
- 3 possibilités
  - Abandon
    - svn revert README.txt
      - abandon des modifications faites dans la copie de travail
  - Ecrassement
    - svn update README.txt
    - cp README.txt.mine README.txt
    - svn resolved README.txt
    - svn commit –m "discard the other revision"
  - Fusion manuelle
    - svn update README.txt
    - edit README.txt
    - svn resolved README.txt
    - svn commit –m "merge the conflicted revisions"
- Remarque
  - En cas de conflit, la commande update crée 3 fichiers
    - `filename.ext.mine` : l'image du fichier de la copie avant l'update
    - `filename.ext.rOLDREV` : le BASE avant l'update
    - `filename.ext.rNEWREV` : le HEAD juste après le commit
    - `filename.ext` contient les marques indiquant les lignes en conflits

# Mode pessimiste (Lock→Modify→Unlock)

- Subversion autorise la pose de verrous exclusifs

- Commandes client
  - svn ps svn:needs-lock '*' README.txt
    - Marque le fichier readonly si un autre rédacteur update avant d'éditer
    - Non obligatoire mais utile !
  - svn lock README.txt -m "pour la release de demain"
  - svn info README.txt
  - edit README.txt
  - svn unlock README.txt

- Commandes administratives
  - Les développeurs (ou les crashes) oublient de déposer leurs verrous
    - svn cleanup
    - svnadmin lslocks c:\repository
    - svnadmin rmlocks c:\repository /README.txt
    - svn info file:///c:/repository/README.txt

# Création et application d'un patch

- Motivations
  - Analyser les différences entre sa copie et une révision (HEAD en général)
  - Analyser les différences entre 2 révisions
  - Transmettre les différences à un co-rédacteur, commiter, …
  - Attacher les différences à une issue de correction (JIRA, …)
- Création d'un patch
  - svn diff README.txt >> r.patch
    - entre le HEAD et sa copie de travail
  - svn diff -r 2:3 README.txt >> r23.patch
    - entre les révisions 2 et 3
  - svn diff --diff-cmd /usr/bin/diff --extension "-i -b" README.txt >> r.patch
    - Utilisation d'une commande de *diff* externe
- Application d'un patch (sur sa copie de travail ou autre)
  - Sur Unix et CygWin, commandes *diff* et *diff3*
  - Sur Windows, ExamDiff, KDiff3, WinMerge, Araxis compare

# Substitution de mots-clé

- Motivations
  - Insérer dans le fichier des informations relatives aux révisions
    - Mise a jour des informations lors des updates
- Méthode
  - Propriété svn:keywords
    - Spécifie la liste de mots clé à substituer
  - Mots-clé substituables
    - Revision (ou Rev): dernière révision ayant changé le fichier
    - HeadURL: URL de la dernière version du fichier
    - Date (ou LastChangedDate): date de la dernière révision ayant changé le fichier
    - Author: auteur ayant changé en dernier le fichier
    - Id: identifiant composite (file rev date author)
- Exemple
  - echo $Id$ $HeadURL$ $Rev$ $Author$ $Date$ >> README.txt
  - type README.txt
  - svn propset svn:keywords "Id Rev HeadURL Date Author" README.txt
  - svn commit README.txt
  - svn update README.txt
  - type README.txt

# Propriétés

- Meta-donnés attachées aux fichiers/répertoires
  - Paires <clé alphanum, valeur ASCII ou binaire>
  - Sémantique définie par les rédacteurs du dépôt ou spécifiques à svn (svn:)
- Propriétés spéciales svn:
    - svn:ignore : liste des patterns de fichiers non versionnables
    - svn:externals : précise que le répertoire provient d'un autre dépôt
    - svn:mime-type : spécifie le type MIME du fichier
    - svn:needs-lock : le fichier doit être verrouillé avant modification
  - Non versionnable
    - svn:log, svn:revision, svn:author, svn:date, svn:autoversioned
- Exemples
  - svn proplist Main.java # liste les propriétés courantes
  - svn propget copyright Main.java # valeur courante de la propriété copyright
  - svn propdel contributors Main.java # suppression d'une propriété
  - svn propset copyright "(c) 2007 Didier Donsez" *.java
  - svn propset license -F /path/to/LICENSE.txt *.java
  - svn propedit copyright *.java # lance l'éditeur %SVN_EDITOR% par défaut
  - svn propedit svn:log –r10 –revprop # édition de l'entrée 10 de l'historique

# Structure type d'un projet

- Subversion ne comprends pas la notion de tags ou de branches
  - → Les tags et les branches peuvent se crées à partir du trunk (ou un autre tag) dans une certaine révision avec la commande copy
    - svn copy project1/trunk project1/tags/release-1.0 --revision 234

- Structure de dépôt recommandée

```
project1/trunk/
project1/tags/
project1/tags/release-1.0/
project1/tags/release-1.1/
project1/tags/release-2.0/
project1/branches/
project1/branches/java7/
project1/branches/aop/
project1/branches/vendor/
project2/
project2/trunk/
…
sandbox/
sandbox/didier
…
```

- Remarque
  - Les tags doivent être read-only ! (hooks)
  - Subversion utilise une *copie retardée* afin de limiter le coût des copies

# Dépôt composite

- Motivation
  - Organiser un dépôt dont certains répertoires correspondent à des répertoires provenant des dépôts externes
    - Utile pour le dév de plugin en tierce partie, …
- Propriété svn:externals (multi-lignes)
  - Spécifie la liste des dépôts annexes/externes
    - svn propset svn:externals "contrib http://svn.other.com/repos/contribs@r123" project1
    - svn propget svn:externals project1
    - svn checkout http://svn.example.com/repos/project1
      - Récupère les fichiers du dépôt principal et des dépôts externes

# TODO
# Bonnes pratiques de la gestion de version

- Commit logical changesets
- Commit early, commit often
- Trunk should be *stable*
- Use a sane repository layout
- Use the issue-tracker wisely
- Track merges manually
  - *When committing the result of a merge, write a descriptive log*
- …
- A lire
  - http://svn.collab.net/repos/svn/trunk/doc/user/svn-best-practices.html

# Dump et chargement de dépôts

- Création d'un dump
  - svnadmin dump repository > repository.svndump

- Création d'un nouveau dépot partiel
  et chargement à partir du dump
  - svndumpfilter include hello/trunk --drop-empty-revs --renumber-revs
    < repository.svndump > hello-trunk.svndump
  - svnadmin create hellorepo
  - svnadmin load hellorepo < hello-trunk.svndump

# Administration d'un dépôt

- Configuration
  - svnserve
    - Deamon/service/standalone utilisant un protocole propriétaire (port=3690)
  - Apache 2.0 + mod_dav_svn
    - Mod Apache 2.0 basé sur WebDAV/DeltaV (RFC 3253)
      - Support de l'auto-versionning
- Sécurité
  - Autorisation
    - username+password en clair/SSH, Certificat SSL
  - Contrôle d'accès
    - Par répertoire (fichier ./conf/authz)
- Hooks (dans le répertoire repo/hook)
  - Scripts (sh,pl,py,bat,…) exécutés par le serveur à certaines étapes de la transaction
    - Contrôle d'accès fin (création de tags, …) , notification (mail,IM) …

# Outils

- Clients et plugin IDE
  - TortoiseSVN, Subclipse, …
  - Taches Ant, plugin Maven SCM, plugin MS VSS, …
- Clients WebDAV et WebDAV/DeltaV
  - Support de la révision
- Navigation Web du dépôt SVN
  - ViewCVS, WebSVN, …
- Activité du SVN
  - Statistiques http://wiki.statsvn.org/
- Migration
  - Convertisseurs CVS vers SVN (http://cvs2svn.tigris.org/)
- Liaisons langage
  - Java (JNI), Perl, Python, C++, C# …
    - SVNKit

# Misc

- Tache ANT **<svn>**

  *http://subclipse.tigris.org/svnant/svn.html*

  - Commandes
    - add createRepository import move status cat delete keywordsset propdel switch checkout diff keywordsadd propget update commit export keywords remove propset copy ignore mkdir revert
  - Exemple

```
<svn javahl="${javahl}" username="guest" password="">
    <checkout url="http://subclipse.tigris.org/svn/subclipse/trunk/svnant/" revision="HEAD"
    destPath="svnant" />
</svn>
<svn javahl="${javahl}">
    <diff oldPath="workingcopy/diffTest/file.txt" outFile="workingcopy/diffTest/patch.txt"/>
</svn>
```

# Plugin SCM Maven http://maven.apache.org/scm/

- Plugin offrant une API commun vers les principaux SCM
  - Commandes
    - Changelog - command to show the source code revisions
    - Checkin - command for commiting changes
    - Checkout - command for getting the source code
    - Diff - command for showing the difference of the working copy with the remote ones
    - Edit - command for starting edit on the working copy
    - Status - command for showing the scm status of the working copy
    - Tag - command for tagging the certain revision
    - UnEdit - command for to stop editing the working copy
    - Update - command for updating the working with the latest changes
    - Validate - validates the scm information on the pom
  - Supported SCM
    - Subversion, CVS, Starteam, Clearcase, Perforce, bazaar
  - URL SCM
    - scm:<scm_provider><delimiter><provider_specific_part>
  - Client
    - java -jar target\maven-scm-client-1.1-jar-with-dependencies.jar checkout c:\temp\maven-scm-client scm:svn:http://svn.apache.org/repos/asf/maven/scm/trunk/maven-scm-client

# SVNKit (http://svnkit.com/)

- pure Java Subversion library (**Open source**)
- enables to Java programs (IDE, WWW document servers, …) to interact with a SVN repository
- Example

```
File dstPath = new File("c:/svnkit");
SVNURL url = SVNURL.parseURIEncoded
      ("http://svn.svnkit.com/repos/svnkit/branches/1.1.x/");
SVNClientManager cm = SVNClientManager.newInstance();
SVNUpdateClient uc = cm.getUpdateClient();
uc.doCheckout(url, dstPath, SVNRevision.UNDEFINED, SVNRevision.HEAD,
    true);
...
uc.doUpdate(dstPath, SVNRevision.HEAD, true);
...
SVNCommitClient cc = cm.getCommitClient();
cc.doCommit(new File[] {new File(dstPath, "www")}, false, "message", false,
    true);
```

# Intégration continue

- Principe
  - Vérifier si les modifications de code n'entraine pas une régression de l'application en cours de développement
- Processus « périodique »
  - A chaque révision ou bien tous les X, …
  - SVN checkout → build → unit testing → …
- Serveurs
  - Apache Continuum
  - Hudson
  - …
  - http://en.wikipedia.org/wiki/Continuous_integration#Software

# Bibliographie

- Page du projet Subversion
  - *http://subversion.tigris.org*
- Livres
  - Ben Collins-Sussman, Brian W. Fitzpatrick & C. Michael Pilato, *Version Control with Subversion*, Pub. O'Reilly, 2004, ISBN 0596004486
    - Version en ligne http://svnbook.red-bean.com/nightly/en/svn-book.pdf
  - Garrett Rooney; *Practical Subversion*; Apress; ISBN 1-59059-290-5 (1st edition, paperback, 2005)
  - Mike Mason; *Pragmatic Version Control Using Subversion*; Pragmatic Bookshelf; ISBN 0-9745140-6-3 (1st edition, paperback, 2005)
  - William Nagel; *Subversion Version Control: Using the Subversion Version Control System in Development Projects*; Prentice Hall; ISBN 0-13-185518-2 (1st edition, paperback, 2005)

# Bibliographie

- Wikipedia
  - http://en.wikipedia.org/wiki/Subversion_%28software%29
- Tutoriel
  - http://eiffelsoftware.origo.ethz.ch/index.php/Subversion_Tutorial
- Memo utile
  - http://www.cs.put.poznan.pl/csobaniec/Papers/svn-refcard.pdf
  - http://www.collab.net/community/subversion/articles/CollabNet_SVNQuickReferenceCard.pdf