

Contents

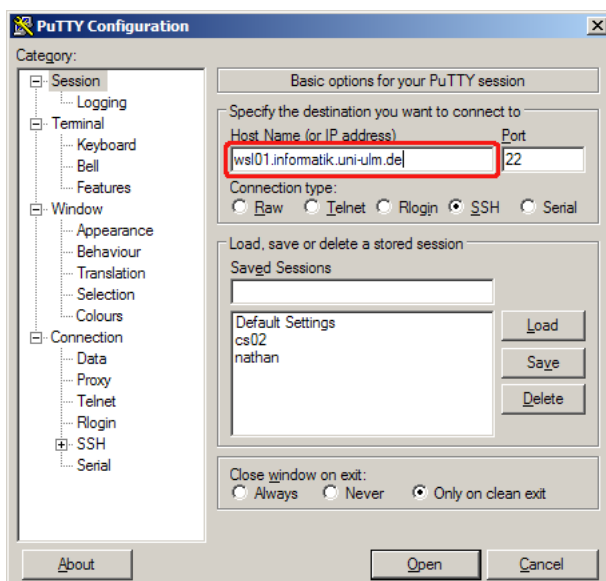
Subversion Einrichten	1
Vorbereitung	1
Public & Private key generieren	1
PuTTY konfigurieren	2
Repository einrichten	3
Subversion konfigurieren	3
Arbeiten mit TortoiseSVN	4

Subversion Einrichten

Vorbereitung

Folgende Tools werden benötigt. Ladet diese herunter und befolgt die Installationsanweisungen.

1. Subversion Client:
<http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91>
2. TortoiseSVN:
<http://tortoisesvn.net/downloads>
3. PuTTY & PuTTYgen:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/>
4. WinSCP:
<http://winscp.net/eng/download.php>



Public & Private key generieren

Startet PuTTY und logt euch mit eurem Pool-Account auf einer der Linux-Workstations ein. wsl01-wsl05 eignen sich am Besten, da diese niemals abgeschaltet werden.

Gebt nun im Prompt folgenden Befehl ein:

```
ssh-keygen -b 1024 -t rsa -N "" -f mykey
```

Es wurden die Dateien mykey.pub als öffentlicher und mykey als private key erstellt.

Schaut nach ob das `.ssh`-Verzeichnis existiert:

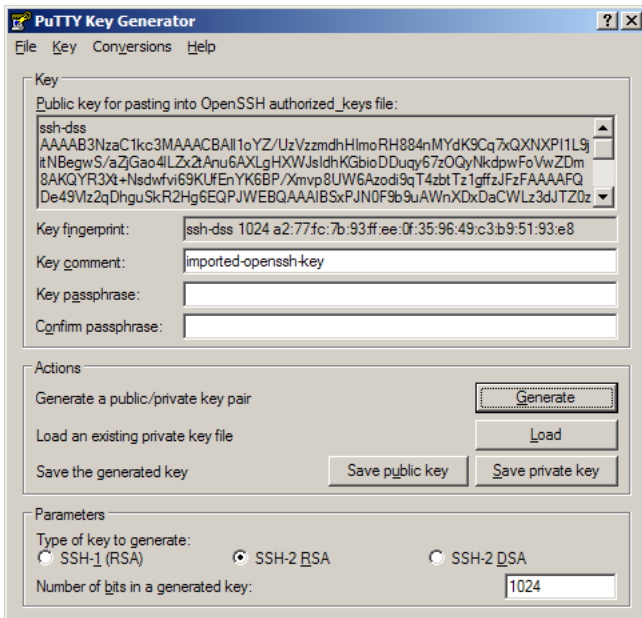
```
dir .ssh
```

Solte dieses nicht existieren, legt es an:

```
mkdir .ssh
```

Wenn ihr drin seid führt folgenden Befehl aus:

```
cp mykey.pub .ssh/authorized_keys
```

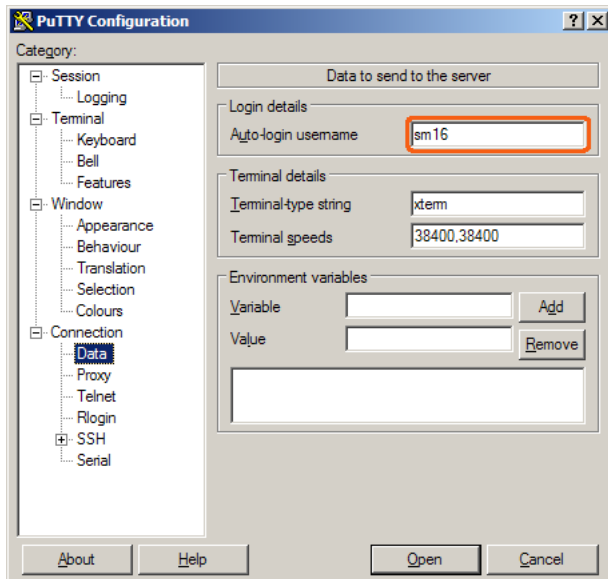


Auf dem Server seid ihr nun fertig und könnt euch ausloggen: `logout`

Als nächstes müsst ihr euch noch den private key mykey herunterladen. Nehmt dazu z.B. WinSCP, das geht am Einfachsten. Logt euch dazu wie vorhin auch mit eurem Pool-Account ein.

Startet PuTTYgen und wählt vom den Menüeintrag `Conversions > Import key` und öffnet die heruntergeladene mykey-Datei.

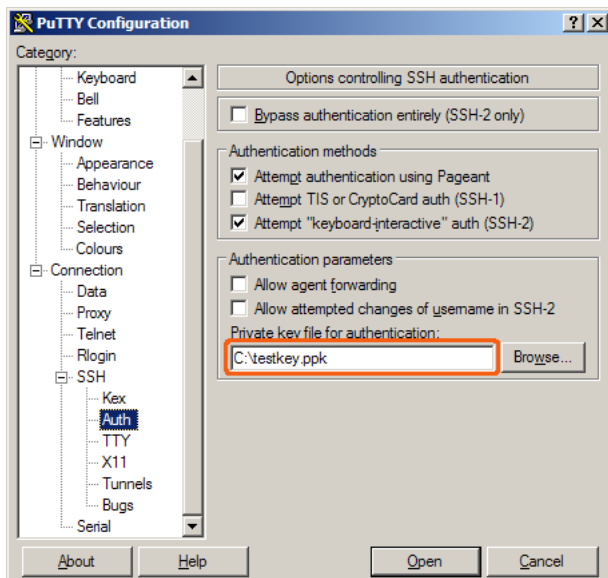
Speichert nun den importierten Schlüssel mittels `File > Save private key als` z.B. `testkey.ppk`. Ignoriert die Warnung, indem ihr aus "Ja" klickt. Die mykey-Datei könnt ihr anschließend löschen.



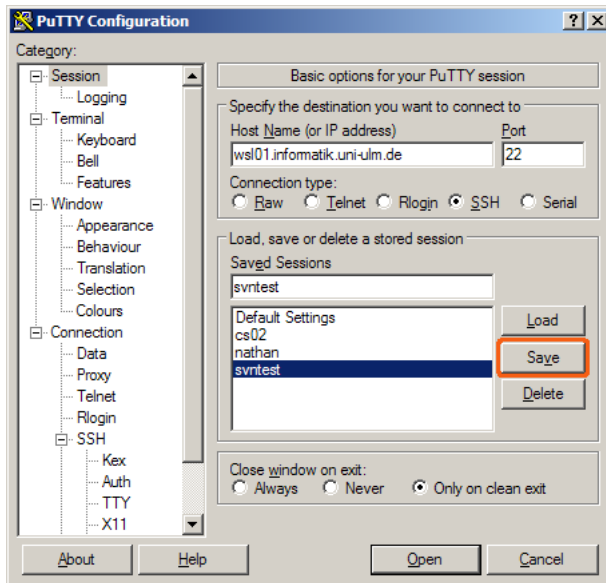
PuTTY konfigurieren

Startet nun wieder PuTTY und tragt im Textfeld Host wieder eine der Workstations `wsl01-wsl05` ein.

Geht dann in die Rubrik `Connection > Data` und tragt im markierten Feld euer Account-Kürzel ein.



Danach setzt ihr unter `Connection > SSH > Auth` im entsprechenden Feld den vollständigen Pfad zu eurem erzeugten ppk-Schlüssel an. Der Einfachheit halber liegt dieser hier direkt in `C:\`.



Abschließend geht ihr wieder zum `Session`-Dialog und speichert eure Einstellungen unter einem sinnvoll benannten Profil ab.

Sinn und Zweck des Ganzen ist es, dass ihr euch in Zukunft bequem und ohne lästige Passwortabfragen einloggen könnt.

Repository einrichten

Jetzt könnt ihr gleich eure gespeicherte PuTTY-Session ausprobieren. Startet PuTTY und ladet euer Profil logt euch ein. Wenn alles glatt geht seid ihr direkt eingeloggt.

Wechselt nun in euer Teamlaufwerk:

```
cd /import/pgm/<TeamName>
```

Führt dort folgenden Befehl aus:

```
svnadmin create <RepositoryName>
```

<RepositoryName> ist dabei ein sinnvoller Name, z.B. `sopra_smile`

Damit ist die Einrichtung auch schon abgeschlossen.

Hinweise:

Achtet darauf dass ihr `"/` oder `\"` statt des normalen `\` verwendet, wenn ihr den Pfad angebt.

Sollte der Pfad Leerzeichen enthalten wie es bei

```
C:/Program Files/TortoiseSVN
```

der Fall ist, so muss der komplette Pfad in Anführungszeichen gesetzt werden.

Also:

```
ssh = "C:/Program Files/TortoiseSVN/bin/TortoisePlink.exe"
```

Subversion konfigurieren

Als letztes müsst ihr die Subversion-Konfiguration noch leicht anpassen. Geht dazu auf `Ausführen` bzw. `Run` im Startmenü und gebt folgenden Befehl ein:

```
notepad %APPDATA%\Subversion\config
```

Geht zum Abschnitt der mit folgenden Zeilen beginnt:

```
### Section for configuring tunnel agents.
[tunnels]
```

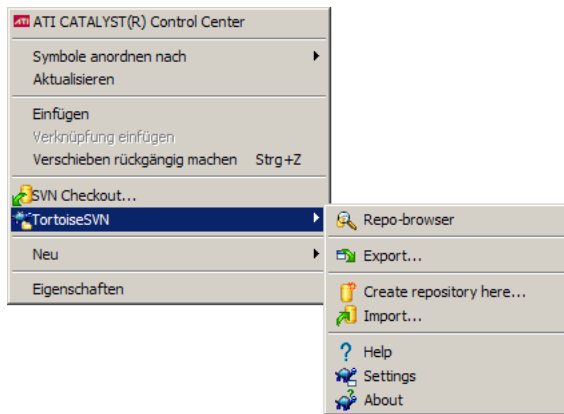
Weiter unten befindet sich die Zeile:

```
# ssh = $SVN_SSH ssh
```

Ersetzt diese durch:

```
ssh = <TortoiseSVN>/bin/TortoisePlink.exe
```

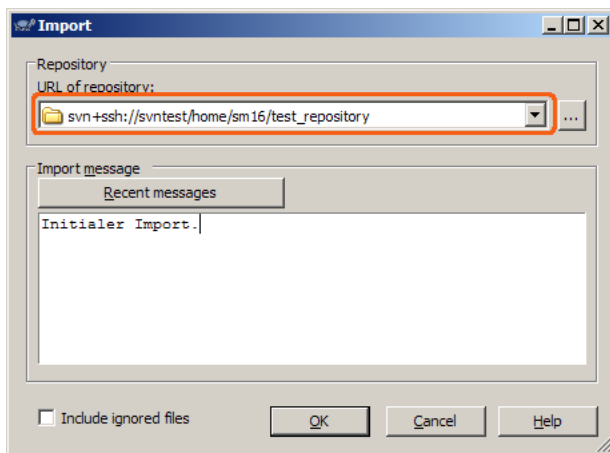
Dabei bezeichnet `<TortoiseSVN>` das Verzeichnis in welches ihr TortoiseSVN installiert habt. Standardmäßig ist dieses `C:/Programme/TortoiseSVN` bzw. `C:/Program Files/TortoiseSVN`



Arbeiten mit TortoiseSVN

TortoiseSVN ist eine Shell-Erweiterung für den Windows Explorer und wird über das Kontextmenü von Dateien und Ordnern bedient.

Markiert nun den Projekt-Ordner den ihr einchecken wollt und wählt über das Kontextmenü TortoiseSVN > Import aus.



Gibt im markierten Feld den Pfad zu dem Repository an, welches ihr eben angelegt habt:

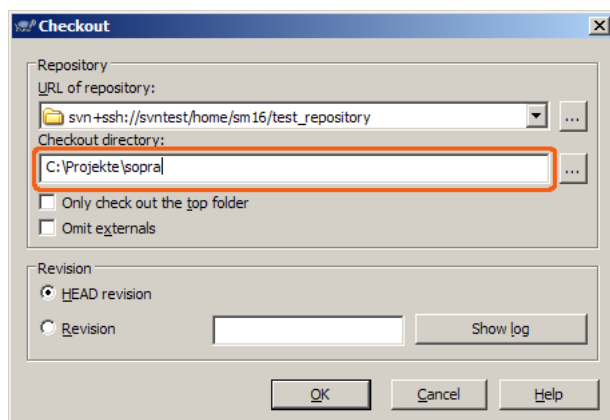
`svn+ssh://<Session>/import/pgm/<TeamName>`

<Session> ist dabei der Name eurer gespeicherten PuTTY-Session, <TeamName> der eures Team-Laufwerks.

Gebt noch eine sinnvolle Log-Message mit an, die beschreibt was ihr gemacht habt. In diesem Fall macht z.B. "Initialer Import." Sinn.

Klickt auf OK und die Dateien werden hochgeladen.

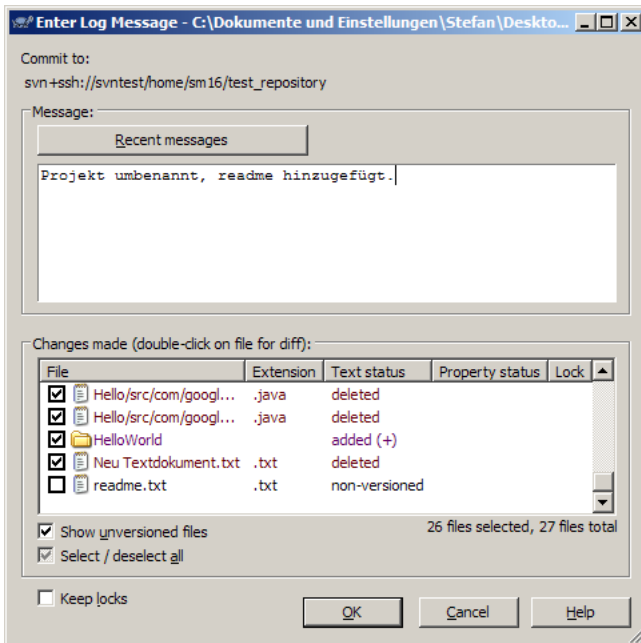
Der Import muss nur einmal zu Beginn und nur von einer Person durchgeführt werden. Euer bisheriges Projekt wird damit erst unter Versions-Kontrolle gestellt.



Sobald euer Projekt im Repository liegt kann dieses ausgecheckt werden. Wählt dazu einfach den Menüpunkt SVN Checkout... aus dem Kontextmenü aus.

Gebt für das Repository den selben Pfad wie eben ein. In das markierte Feld gebt ihr das Verzeichnis in welchem die lokale Arbeitskopie abgelegt werden soll.

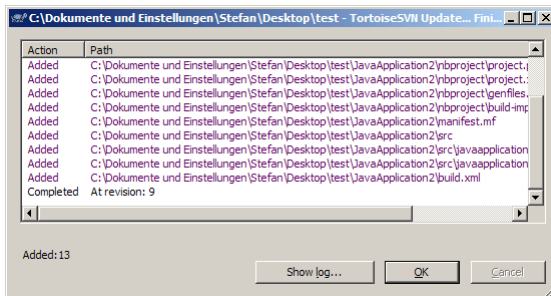
Klickt auf OK und das Projekt wird heruntergeladen.



Nun könnt ihr lokal am Projekt arbeiten. Wenn ihr fertig seid und eure Änderungen gespeichert habt müsst ihr ein Commit durchführen. Klickt dazu auf euren Projektordner oder einzelne Dateien und wählt im Kontextmenü SVN Commit... aus.

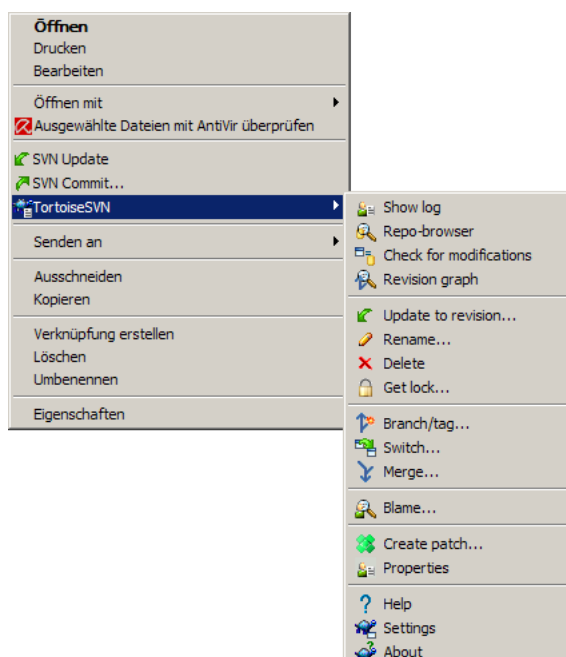
Bereits vorhandene Dateien werden geändert, neue bzw. gelöschte Dateien werden im Repository hinzugefügt bzw. entfernt.

Bevor ihr allerdings ein Commit durchführt solltet ihr auf jeden Fall sicherstellen, dass das Projekt fehlerfrei ist und kompiliert.



Bevor ihr aber am Projekt Änderungen vornehmt solltet ihr sicherstellen, dass eure Version aktuell ist. Damit geht ihr von vornherein eventuellen Konflikten aus dem weg.

Führt dazu ein Update aus. Sämtliche Änderungen anderer Team-Mitglieder werden dabei übernommen. Sollten dabei noch bisher ungelöste Konflikte vorliegen, können diese mit dem integrierten Diff-Tool aufgelöst werden.



Solltet ihr nachträglich Dateien löschen oder umbenennen wollen macht das nicht direkt im Explorer, da sonst Subversion ins Straucheln kommt. Wählt stattdessen die entsprechenden Funktionen aus dem TortoiseSVN-Menü.

Bei den ersten vier Einträgen handelt es sich um hilfreiche Tools mit denen man

- Log-Messages,
- bisher durchgeführte Aktionen,
- die Repository-Struktur,
- etc.

durchsuchen und visualisieren kann.