Zugang zum SVN-Repository

Der Zugang zum SVN-Repository erfolgt über eine SSH Verbindung. Die Authentifizierung erfolgt mit public-key authentication. Dafür benötigt man SSH-Keys. Wie das alles funktioniert ist hier genau beschrieben.
Für einen Zugang zum Server schreibt uns eine E-Mail mit eurem gewünschten Usernamen, Vor- und Nachname an idefix at fechner.net. Ganz wichtig, schreibt rein was ihr machen wollt, damit wir die Arbeit koordinieren können und Sachen nicht doppelt gemacht werden. Wir geben einen Zugang nur an Leute raus, die wir kennen und die wirklich mithelfen wollen (also vorher unbedingt im IRC melden).

Für diejenigen, die im Umgang mit Subversion schon geübt sind, hier die Adresse unter dem das Repository ausgecheckt werden kann: *svn+ssh://<user>@dev.freebus.org/usr/local/svn/freebus*
Im folgenden wird beschrieben, welche Schritte und Tools nötig sind um den Zugang einzurichten. Unter Punkt 1 für Windows, Punkt 2 für Linux.
Der SSH-Fingerprint unseres Servers ist: 53:06:27:da:4d:b6:5d:53:52:da:0b:46:a3:00:33:ee

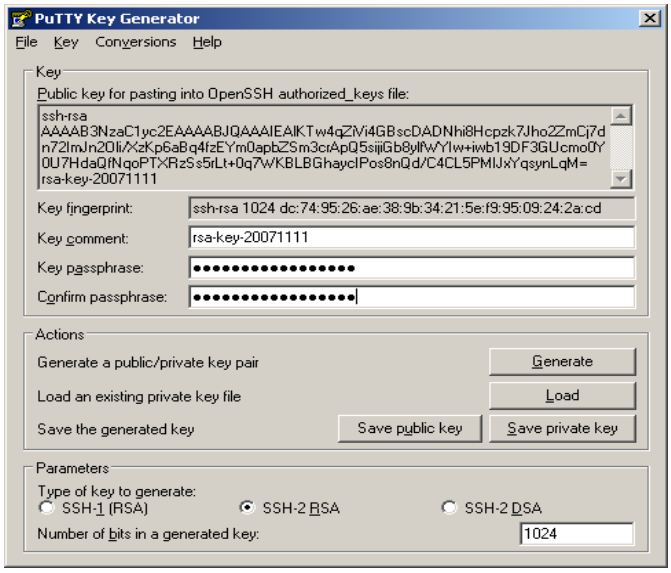
1. WINDOWS

1.1. SSH-Keys anlegen

Jetzt werden ein paar Programme benötigt.

Ssh-Key Maker: hier

Dann den Keymaker starten und auf Generate klicken.



Bei Key passphrase eine langes Passwort eingeben, mit dem der SSH Schlüssel verschlüsselt werde soll, sich dieses Passwort gut merken und unbedingt ein sicheres wählen, dieses Passwort muss nur einmal eingegeben werden, bei jedem Rechner Neustart. Bei Confirm passphrase Passwort nochmal wiederholen.

Dann auf Save private Key klicken und unter einem sicheren Verzeichnis speichern, ACHTUNG diese Datei auf KEINEN Fall weitergeben!

Nun kopiert ihr euch den public key, der ganz oben in der grossen Textbox steht (Markieren und Strg+c), fängt mit ssh-rsa an. Jetzt notepad aufmachen (Start->Ausführen: notepad) und CTRL+v drücken, jetzt sollte da eine lange Zeile stehen. Jetzt auf speichern gehen, bei Dateityp Alle Dateien wählen, bei Dateiname authorized_keys eingeben, Achtung keine Dateiendung! Falls notepad es mit der Endung .txt gespeichert hat, die Datei umbennen und .txt entfernen.

1.2. SSH-Key laden

Jetzt brauchen wir das Programm SSH-Key Agent.

Das Programm starten jetzt habt ihr unten rechts in der Menüleiste das Icon:

Rechtsklick drauf Add Key und jetzt den Key laden den ihr vorher erstellt habt. Jetzt die Passphrase eingeben die ihr vorher eingegeben habt.

1.3. SSH-Key hochladen

Jetzt brauchen wir das Programm WinSCP.

Das Programm installieren und starten.

Jetzt eine neue Verbindung zu:

Host: dev.freebus.org

User: euren Usernamen den ihr bekommen habt, kein Passwort eingeben!

Dann auf Login. Jetzt solltet ihr eine Abfrage wegen dem SSH-Fingerprint bekommen, vergleicht, das der stimmt (siehe dieses Dokument ganz oben). Anschliessend das Passwort eingeben, das ihr bekommen habt.

Jetzt rechts ein Verzeichnis anlegen: .ssh (Achtung der . ist wichtig)

In das Verzeichnis wechseln und dort die vorher angelegt authorized_keys da reinkopieren.

Danach das Programm beenden, nochmal neu starten und nochmal wie oben beschrieben einloggen, jetzt sollte WinSCP nicht mehr nach einem Passwort fragen, falls doch, habt ihr einen Fehler gemacht. Dann prüft nochmal alles Schritt für Schritt:

1. Name der Datei muss authorized_keys sein, keine Endung
2. Datei auf den Server kopiert in das Verzeichnis .ssh

3. pageant gestartet und Key geladen

1.4. SVN-Repository auschecken

Jetzt holt euch das Programm SVN-Tortoise. Das Programm installieren.

Dann ein Verzeichnis auf der Platte anlegen, bei mir heisst es freebus. Dort reingehen, Rechtsklick und SVN-Checkout auswählen.

Bei URL das folgende eingegeben:

svn+ssh://<user>@dev.freebus.org:/usr/local/svn/freebus

<user> durch euren Usernamen ersetzen

Dann auf OK klicken, jetzt sollte er denn Quellcode auschecken.

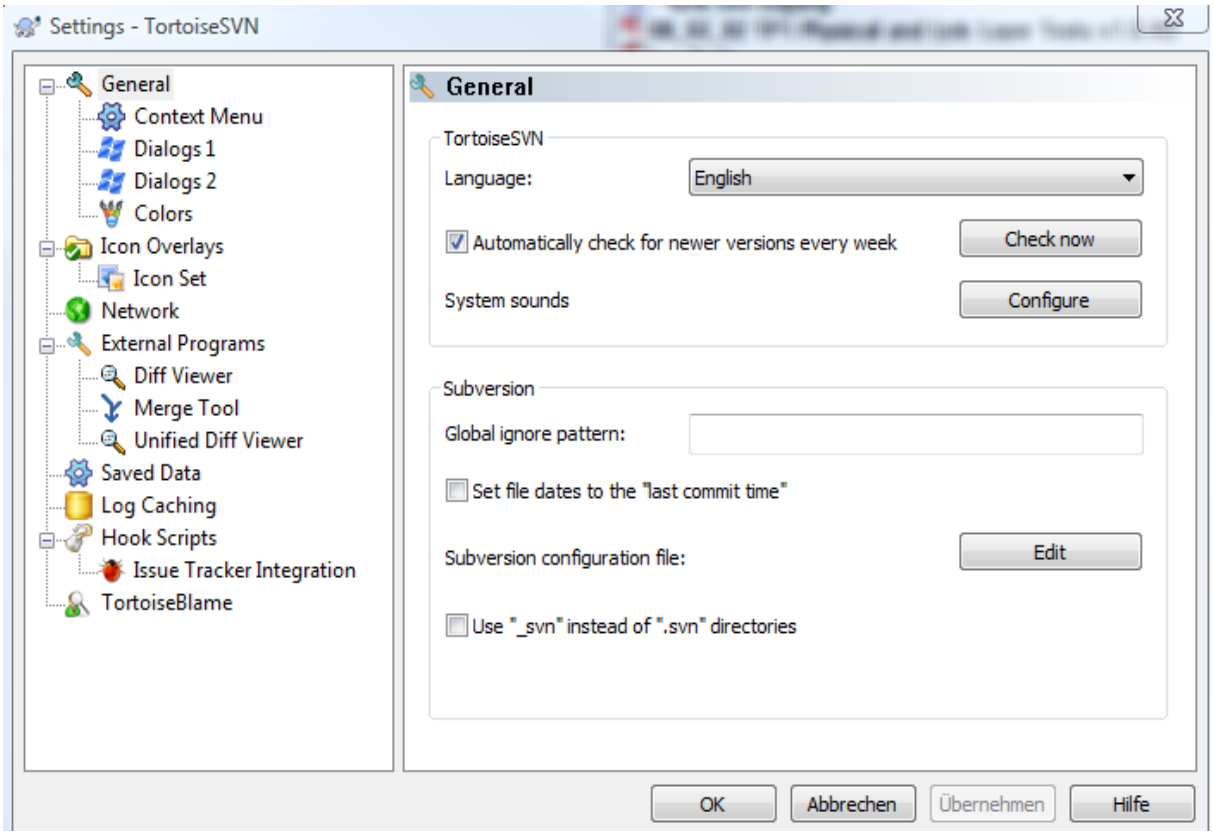
Jetzt könnt ihr am Quellcode arbeiten und diesen wieder einchecken.

1.5. Ganz normal Arbeiten mit SVN

Wenn alles mal eingerichtet worden ist und der Rechner neu gestartet worden ist, einfach nochmal das machen was bei SSH-Key laden beschrieben ist.

1.6. Tortoise-SVN konfigurieren

Geht in das Repository rein, dann ein Rechtsklick TortoiseSVN->Settings. Dann klickt auf Edit:



Dann unter den passenden Einträgen das folgende eintragen:

```
[miscellany]
enable-auto-props = yes


[auto-props]
*.hex = svn:mime-type=application/octet-stream
*.pdf = svn:mime-type=application/octet-stream
*.c = svn:eol-style=LF
*.h = svn:eol-style=LF
Makefile = svn:eol-style=LF
```

2. LINUX

Für die Verwendung von Subversion unter Linux benötigt man zumindest einen *ssh*- und einen *subversion*-Client. Ersterer ist in der Regel bereits im Standardumfang einer Distribution enthalten, subversion kann üblicherweise aus den Paketquellen nachinstalliert werden. In Ubuntu-Linux heißt das entsprechende Paket beispielsweise *subversion.*

Wer kein Freund der Kommandozeile ist oder mehr Komfort braucht, hier ein paar Links:

- Eclipse bietet über ein plugin SVN integration: http://subclipse.tigris.org/
- Für Ubuntu gibt's ein Paket für die Integration von SVN in Nautilus (nautilus-script-collection-svn)
- Für KDE gibt's KDEsvn http://kdesvn.alwins-world.de/
- Für Gnome gibt's NaughtySVN http://naughtysvn.tigris.org/

2.1. SSH-Keys anlegen

SSH-Keys werden ganz einfach durch die Eingabe von

ssh-keygen

in der Kommandozeile angelegt.

Generating public/private rsa key pair.
Enter file in which to save the key (/home/username/.ssh/id_rsa)

Als erstes wird man nach dem Speicherort des keys gefragt, üblichweise passt der Standardvorschlag, also einfach mit Return bestätigen.

Enter passphrase (empty for no passphrase):

Als nächstes folgt die Eingabe der passphrase, die später zum entsperren des Keys verwendet wird. Sie sollte möglichst sicher gewählt werden (mindestens 8 Zeichen, keine Wörter, am besten Zahlen und Sonderzeichen einbauen und Groß- und Kleinschreibung mischen). Man kann auch keys ohne passphrase anlegen, davon ist aber auf jeden Fall abzuraten.

Nach Bestätigung der passphrase ist man auch schon fertig, es wurde ein öffentlicher und ein privater Schlüssen angelegt.

2.2. SSH-Key laden

Um nicht jedesmal, wenn man eine Operation im subversion repository durchführt, seinen private key entsperren zu müssen, empfielt es sich, ihn einmal mit *ssh-add* zu laden. Danach bleibt er für die bestehende session geladen (evtl. ist es notwendig keychain zu installieren). Wenn man die keys im Standardverzeichnis liegen hat, reicht ein simples

ssh-add
Enter passphrase for /home/username/.ssh/id_rsa:

2.3. SSH-Key hochladen

Um den key mit dem Server bekannt zu machen, muss am Server der Ordner *.ssh* angelegt werden und darin der public(!) key in der Datei *authorized_keys* geschrieben werden. Das geht zB so:

mkdir /tmp/.ssh
cp ~/.ssh/id_rsa.pub /tmp/.ssh/authorized_keys
scp -r /tmp/.ssh <user>@dev.freebus.org:

Überprüft hier, ob der Host-Fingerprint passt, steht ganz oben in diesem Dokument.

2.4. SVN-Repository auschecken

Zum checkout des Repositorys legt man sich am besten einen passenden Ordner an, zB freebus-svn. Darin führt man dann in der Kommandozeile, mit geladenem key, folgendes Kommando durch:

svn co svn+ssh://<user>@dev.freebus.org/usr/local/svn/freebus

2.5. Arbeiten mit SVN

Die Dateien können nun bearbeitet werden und mit *svn ci <Dateiname>* werden die Änderungen committed. Neue Dateien/Ordner kann man mit *svn add <Dateiname>* hinzufügen. Um sich auf den neuesten Stand zu bringen, reicht ein einfaches *svn update* im repository Verzeichnis. Damit kann man auch ungewollte Änderungen rückgängig machen: dazu einfach die Datei lokal löschen und mit *svn update* die aktuelle Version holen.

Für eine ausführliche Einführung in die Arbeit mit svn gibt es jede Menge Online Tutorials.

3. Was ist beim Programmieren zu beachten

Wir nutzen einen Tabstob von 5 Zeichen, das sollte in jeder IDE einstellbar sein (z.B. AVR Studio 4).

Kommentare im Programm werden in Englisch gehalten, genauso die Beschreibung beim einchecken. Wenn Code eingecheckt wird ist IMMER eine Beschreibung zu schreiben!

Die Dokumentation im Quellcode wird mit doxygen erstellt.

So viel Spass beim Mithelfen, wenn es Fragen geben sollte, hilft euch jeder gerne weiter.

Matthias (& Thomas alias thoma5 – Linux Teil)