

Access to SVN-Repository

The access to the SVN repository is only possible with a ssh connection. For the identification the public key mechanism is used. For this a private/public key pair is necessary. Further details are given in that document.

For access to the server write an e-mail with your wished username your last and first name to idefix at fechner.net. Please also mention what you want to do!

That is important because we try to coordinate the team that no double work is done. We provide access to our SVN server only to known people they really want to help us actively in developing or testing our products (so please visit us on the IRC server).

For all the people with advanced svn knowledge the address for the repository is:

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

The SSH-Fingerprint of the server is: 53:06:27:da:4d:b6:5d:53:52:da:0b:46:a3:00:33:ee

Chapter one is for Windows and Chapter 2 for Linux.

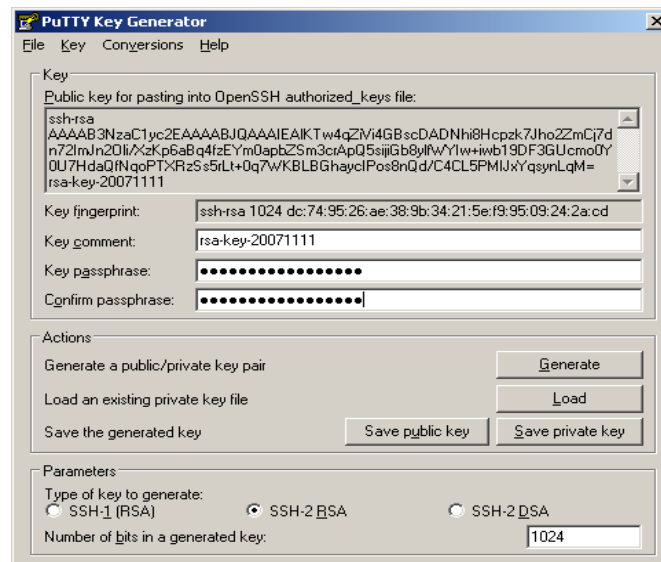
1. WINDOWS

1.1. Create SSH-Keys

At first we need some programs.

Ssh-Key Maker: [here](#)

Now start the keymaker and click Generate.



Use at Key passphrase a long phrase that secures your private SSH Key. You must retype that passphrase only after each reboot if your computer when you load the key pair. Repeat that passphrase at Confirm passphrase.

Then click Save private Key and save the key on a secure place, CAUTION do NOT give that key

to anyone!

Now copy the public key you can see in the top of the window (mark it and Ctrl+c), it starts with ssh-rsa. Open notepad with (Start → Execute: notepad) and press CTRL+v. Now you should have one long line in that file. Now save the file with the filename `authorized_keys`. Notepad always append the file extension `.txt` so you must rename the file in the windows explorer after you saved it!

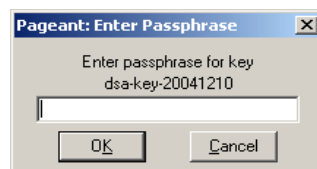
1.2. Load the SSH-Key

Now we need the program [SSH-Key Agent](#).

Start the program. In your windows task bar there should be the following icon:



Right click and Add Key. Select the key we saved before from the key generator. Now you must enter your passphrase you defined before.



1.3. Store SSH-Key on the Server

Now download the program [WinSCP](#).

Install and start the program.

Now create a connection to:

Host: `dev.freebus.org`

User: your username you received by e-mail, don't insert a password!

Click on Login. Know WinSCP displays the SSH-Fingerprint of the server. Compare the fingerprint with the one in that document (see top of this document). Now insert you password you received by email.

Create on the right side a directory: `.ssh` (Caution the `.` is important)

Switch to that directory and copy the first saved `authorized_keys` there.

Close WinSCP and restart it, repeat the login steps above, WinSCP should NOT ask for a password now. If it asks for a password you did something wrong, check the following steps:

1. Name of the file must be `authorized_keys`, no extension
2. The file is stored on the server in the directory `.ssh`
3. pageant is started and the key is loaded

1.4. Checkout the SVN-Repository

Download the program [SVN-Tortoise](#) and install it.

Create a directory on your haddisk called freebus. Go there right click and select SVN-Checkout.

As URL insert:

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

<user> replace it with your username

Click on OK, no it should download the repository.

Now you can work on repository.

1.5. Daily Work with the SVN

When you restart your computer you must reload the SSH-Key like described in chapter 1.2.

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 Schlüssel angelegt.

2.2. SSH-Key laden

Um nicht jedesmal, wenn man eine Operation im subversion repository durchführt, seinen private key entsperren zu müssen, empfiehlt 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 Repositories 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. What You Should Know Before Start

We use a tab stop of 5 characters, you should be able to define that in every IDE (e.g. AVR Studio 4).

The language for comments is English, the same is for descriptions while committing to the repository. You must always define a comment when you do a checkin to the repository.

The documentation in the source code is generated with the help of [doxygen](#).

If you have questions ask in the IRC channel or use the forum or mailing lists. Good luck.

Matthias (& Thomas alias thoma5 – Linux Part)