Inhaltsverzeichnis

1 Zugang zum GIT Repository

Der Zugang zum GIT Repository ist je nach Berechtigung unterschiedlich. Es gibt für jeden einen read-only Zugang, der Link dazu ist im Repository sichtbar, welcher sich auf der Homepage befindet, als Beispiel für das Hardware/Docu Repository ist der Link git://git.freebus.org/freebus.

Einen Schreibzugriff auf das Repository gibt es, wenn einige commits von Form von Diffs zur Verfügung gestellt worden sind. Um einen Schreibzugriff einrichten zu können, ist es notwendig sich einen SSH Key zu generieren und mir den Public Key zuzuschicken. Schickt die email mit dem key bitte an root at freebus Punkt org.

Der Link um dann zu commiten ist:

git@freebus.org:freebus

Falls ihr an einem größeren Projekt arbeitet, bitte ich euch, sich einen Account in unserem Bugtracker anzulegen. Schreibt mir dann an root at freebus.org eine Email, mit einem Projekt Titel, Beschreibung und eurem Account Namen im Bugtracker.

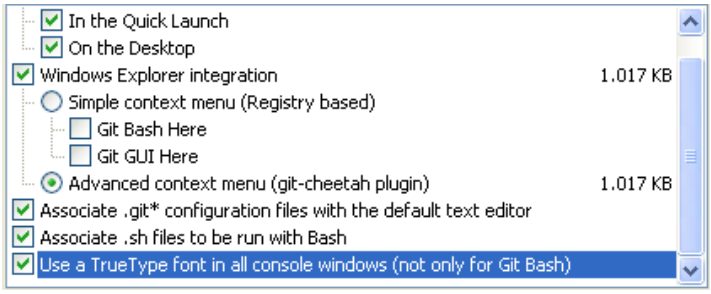
Ich lege euch dann ein Projekt an. Ihr könnt dann alle wichtigen Daten dort hinterlegen, wie Docu, es gibt ein Wiki, es können andere Leute mithelfen, etc.
Wir wollen hier verhindern, das Projekte doppelt gemacht werden und das es leichter ist rauszufinden, wo man mithelfen kann/will.

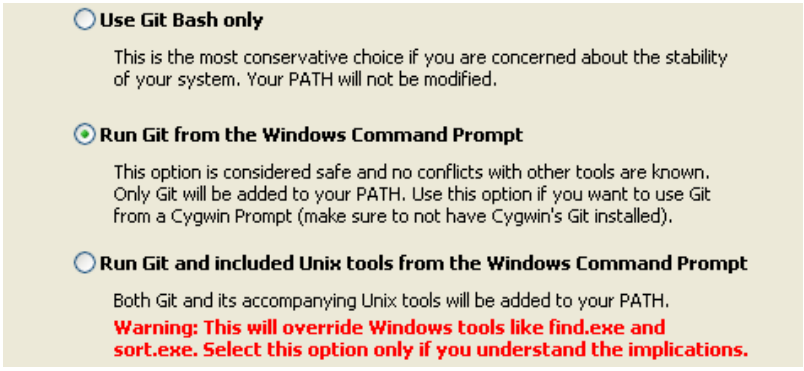
2 Git unter Windows installieren

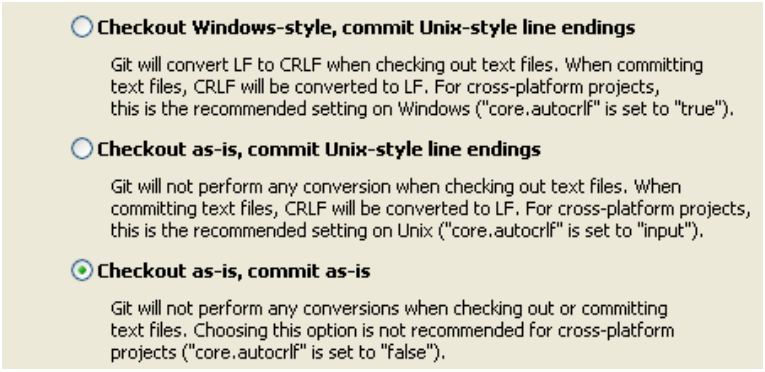
Zuerst müsst ihr euch Git runterladen. Das bekommt ihr hier:

http://git-scm.com/downloads

Bei der Installation ist das folgende auszuwählen:







Danach wird Git auf dem Rechner installiert.

Jetzt installieren wir uns noch eine schöne GUI für Git. Ich selber benutze unter Windows ganz gerne Git Extension:

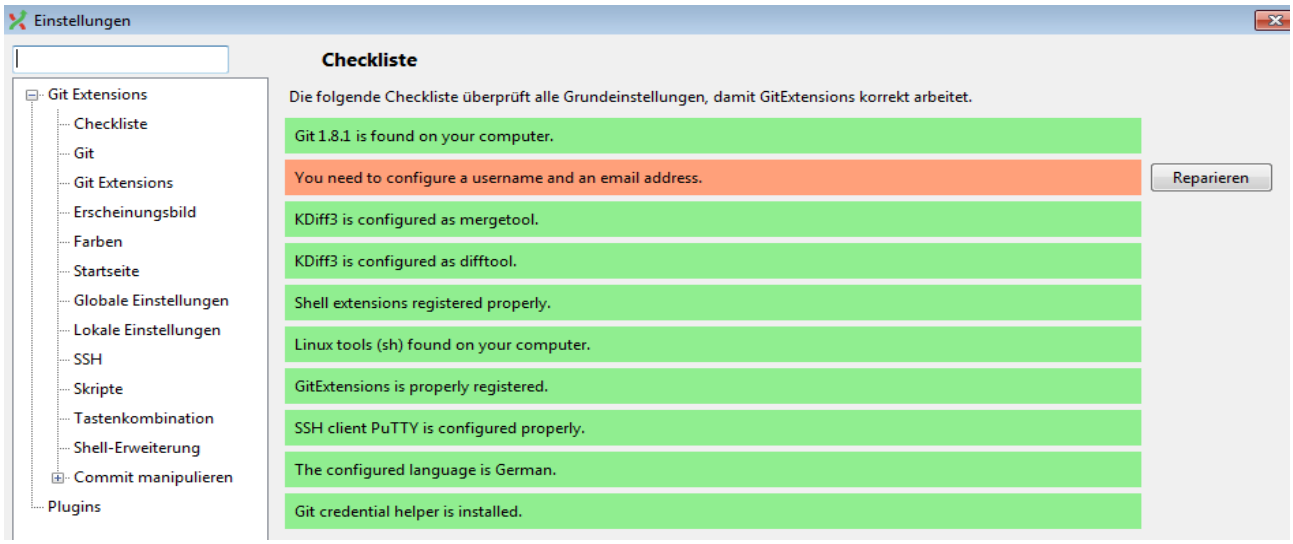
http://code.google.com/p/gitextensions/

Bei der Installation Kdiff bei der Installation auswählen und die Verwendung von Putty (statt OpenSSH). Falls die Installation nicht weiter geht, liegt es wahrscheinlich an Kdiff, das nach ein paar Tastendrücken wartet (einfach mal das Installationsfenster ein bisschen nach rechts schieben).

Falls ihr noch kein .net installiert habt, müsst ihr das vorher machen, ich habe bei mir die Version 4 verwendet (Version 4.5 geht nicht):

http://www.microsoft.com/en-us/download/details.aspx?id=24872

Jetzt müssen wir noch ein paar Einstellungen an Git machen. Hierfür starten wir Git Extensions. Wir sollten jetzt eine Checkliste sehen:



Einfach rechts auf Reparieren klicken und dann seinen vollen Namen und Email eintragen:

Dann auf ok, jetzt sollte Git Extension starten. Jetzt gehen wir auf das Menu Git und starten dort die Git Bash, in der wir noch das folgende eingeben:

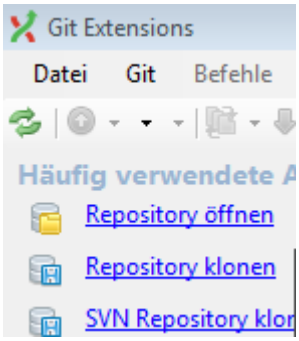
```
git config --global branch.autosetuprebase always
git config --global push.default current
```

Dannach könnt ihr die Git Bash wieder mit exit schliessen.

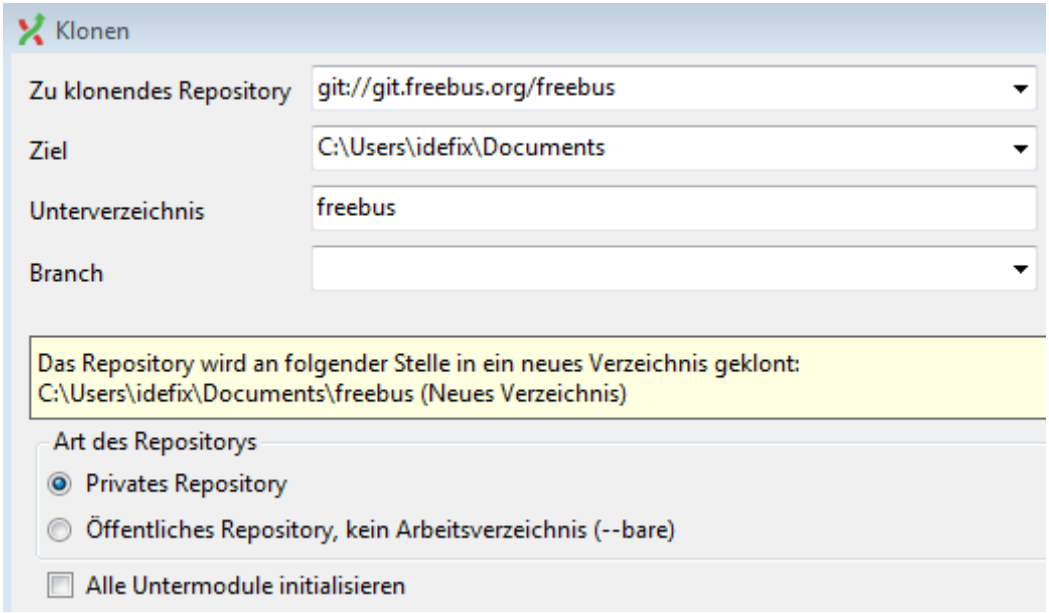
Jetzt haben wir alle Tools installiert um unsere erste Änderung dem Projekt bei zu steueren, ich zeige es hier an einem kleinen Beispiel, wie es geht.

3 Wie kann ich mithelfen

Jetzt starten wir Git Extensions und machen einen Readonly Clone. Hierfür in Git Extensions unter häufig verwendete Aktionen auf Repository klonen klicken:

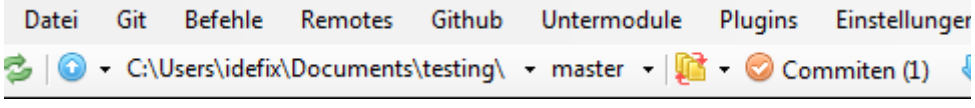


Bei URL tragen ihr die ein, die ihr im Web unter dem Repository findet, an dem ihr etwas beisteuern wollt, z.B.:

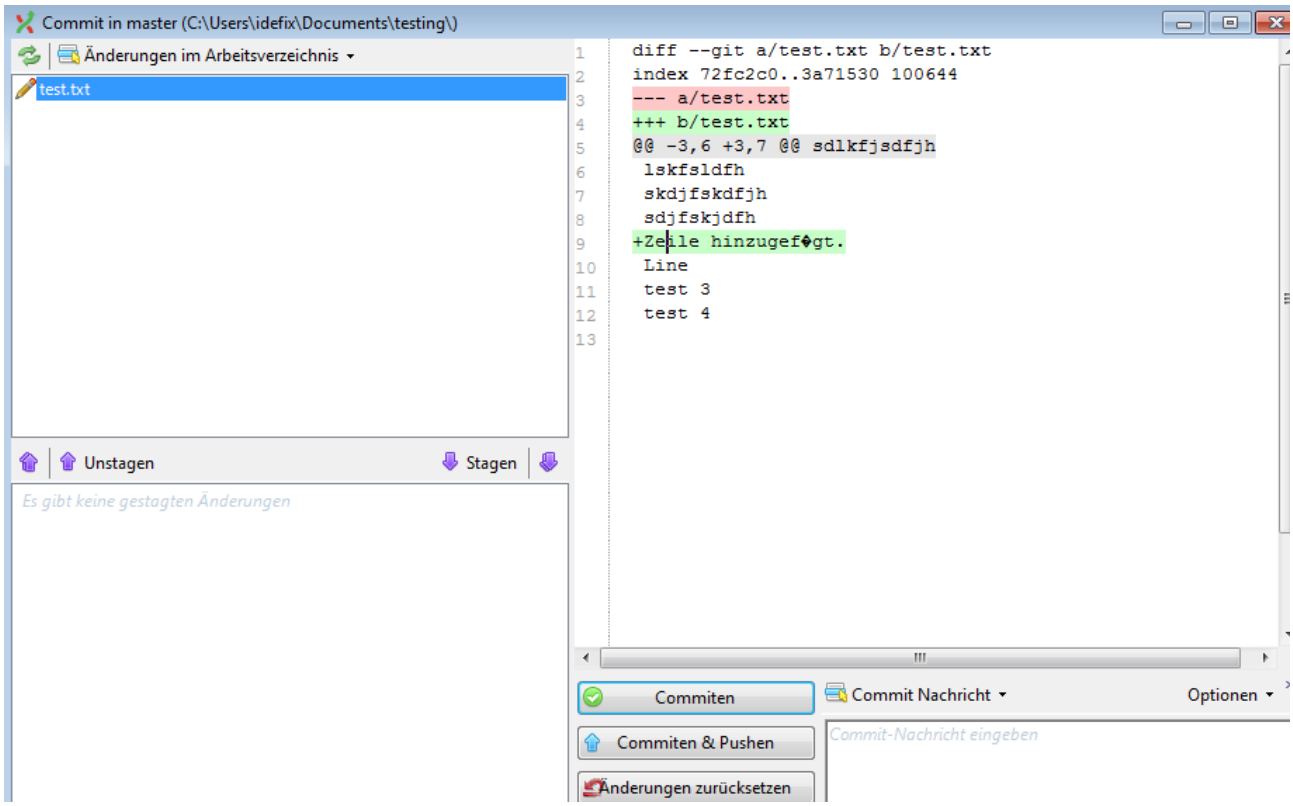


Wenn der clone fertig ist, könnt ihr die Dateien anpassen, neue anlegen, ganz was ihr ändern wollt. Jetzt ist es notwendig die Änderungen in das lokale Repository zu commiten und hier eine entsprechende Beschreibung anzugeben. Als Beschreibung tragt bitte ein welche Funktionalität ihr geändert habt und ganz wichtig wieso, dass hilft euch später selber dann auch weiter. Die Beschreibung kann in Englisch oder Deutsch sein, aber eine englische Beschreibung ist vorzuziehen, da diese auch andere Leute lesen und verstehen können, die kein Deutsch können.

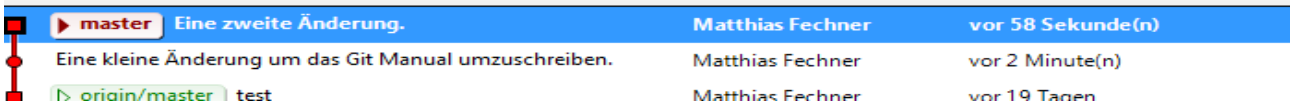
Wenn ihr Daten geändert habt und das Repository in Git Extensions geöffnet ist, seht ihr das oben am Commit Knopf. In diesem Beispiel wurde eine Datei geändert:



Um die Änderungen jetzt zu commiten, einfach auf Commit klicken, dann sieht man schöne was geändert wurde:

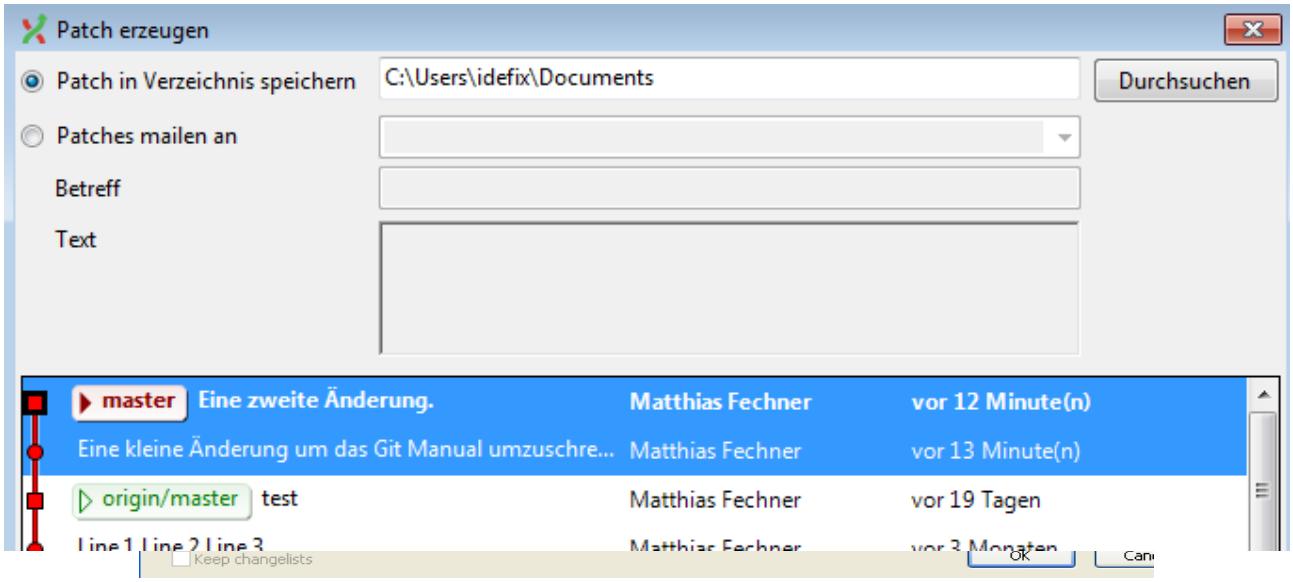


Jetzt kann man rechts eine Zeile markieren und S drücken, damit wird diese Zeile für den nächsten Commit vermerkt (dem Stage-Index hinzugefügt). Wenn man alle Änderungen der aktuellen Datei commiten will, kann man auch links aus Stagen klicken. Alles was im oberen Fenster auftaucht ist noch nicht beim Commit erhalten, alles was unten gelistet wird, wird beim nächsten Commit commited. Wenn wir alles makiert haben, was wir commiten wollen, tragen wir rechts unten noch eine Commit Meldung ein, die erklärt, was wir gemacht haben und wieso wir es gemacht haben (englisch wäre schön, aber nur deutsch ist auch ok). Dann auf Commiten klicken. Es könnte dann in etwa so aussehen:



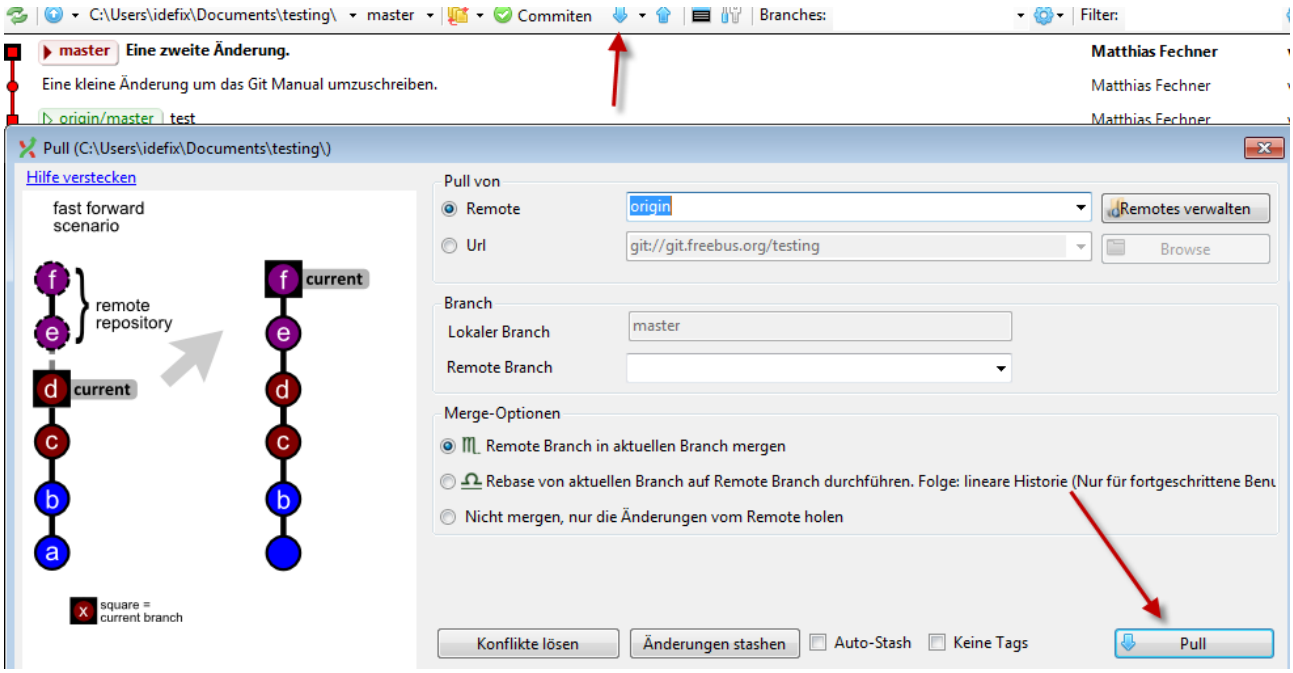
Wir sehen hier, unseren lokalen Zweig (master) und dem offiziellen Zweig (origin/master), welcher unsere beiden Commits noch nicht enthält.

Als nächster Schritt erstellen wir nun eine Patchdatei, welche dann an die entsprechende Mailingliste verschickt werden kann. Hierfür wählen wir im Menu Befehle → Patch erstellen. Wenn man auf dem Rechner ein Emailprogramm installiert hat, kann man die Patches gleich direkt verschicken. Die Patch Dateien können aber auch in einem Verzeichnis gespeichert werden und dann von einem anderen Rechner versendet werden. Jetzt markieren wir die Commits, die wir als Patch einreichen wollen und klicken auf Patch(es) erzeugen.

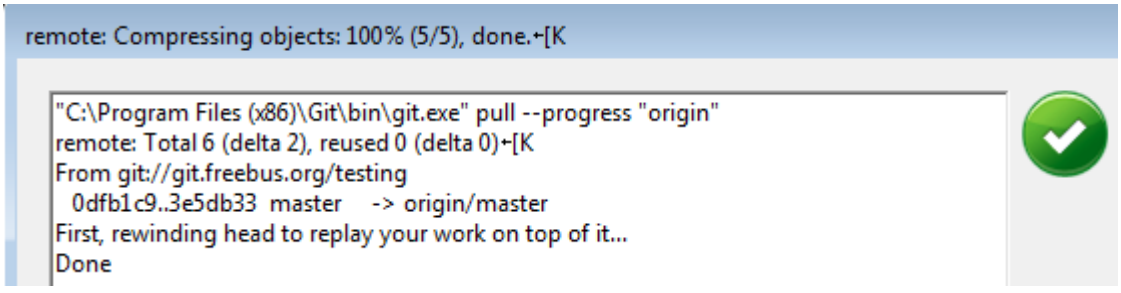


Git Extensions hat nun pro Commit eine Patch Datei erzeugt, diese senden wir an die passende Mailingliste, damit die Änderungen ins Repository gelangen

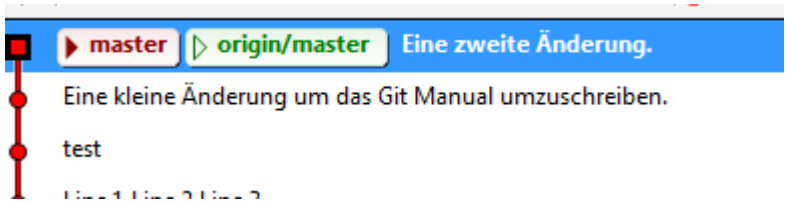
Nachdem unsere Änderung ins offizielle Git geladen wurde, können wir einfach ein Pull machen und sehen dann, das unsere beiden Commits jetzt auf dem Server sind:



Meldung nach dem Pull:



Wenn wir jetzt in die History schauen, sehen wir:



Unsere beiden Commits sind jetzt also auf dem Freebus Server aufgenommen worden.

Nachdem ihr ein paar Commits auf diese Weise dem Projekt beigesteuert habt, richte ich euch gerne einen Schreibzugang für das Repository ein. Wie das geht seht ihr dann in 4Schreibzugriff einrichten.

4 Schreibzugriff einrichten

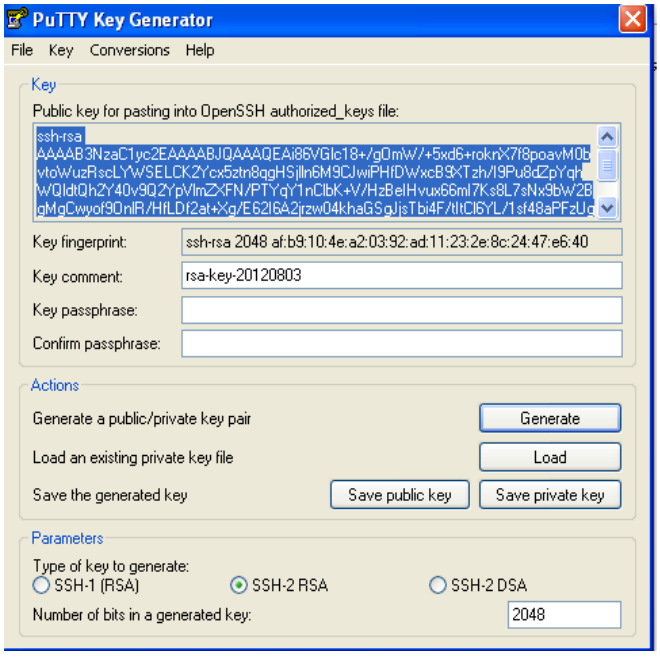
So wird der Schreibzugriff für Windows eingerichtet. Falls ihr schon einen SSH-Key habt, müsst ihr keinen neuen anlegen. Es reicht wenn ihr mir den Public Key wie in 4.1SSH-Key anlegen beschrieben zuschickt.

4.1 SSH-Key anlegen

Jetzt brauchen wir ein paar weitere Tools, welche aber mit Git Extensions schon alle installiert worden sind. Bei einem Windows 7 liegen diese unter:

C:\Program Files (x86)\GitExtensions\PuTTY

Als erstes legen wir uns einen neuen Key an, also starten wir dort das Programm puttygen und klicken in dem Programm auf Generate.



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, somit stört ein langes Passwort nicht. 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! Diesen Key auf jeden Fall mit in euer Backup aufnehmen!

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 gibt euren Usernamen auf der Homepage an, mit der Endung .pub. In meinem Fall wäre das idefix.pub. Falls notepad es mit der Endung .txt gespeichert hat, die Datei umbenennen und .txt entfernen. Diesen Key schickt ihr mir bitte an root at freebus.org.

4.2 SSH Key laden

Jetzt brauchen wir einen Key Agent, in dem wir unseren SSH Key laden, dazu starten wir das Programm pagent in dem gleichen Verzeichnis, aus dem wir auf den Key Generator gestartet haben.

In der Windowsleiste sehen wir nun das Icon:

Rechtsklick drauf Add Key und jetzt den Privat Key laden den ihr vorher erstellt habt.

Jetzt die Passphrase eingeben die ihr vorher eingegeben habt. Jetzt ist der Key geladen und wir können ein Clone machen, mit dem wir dann auch Änderungen zum Server pushen können.

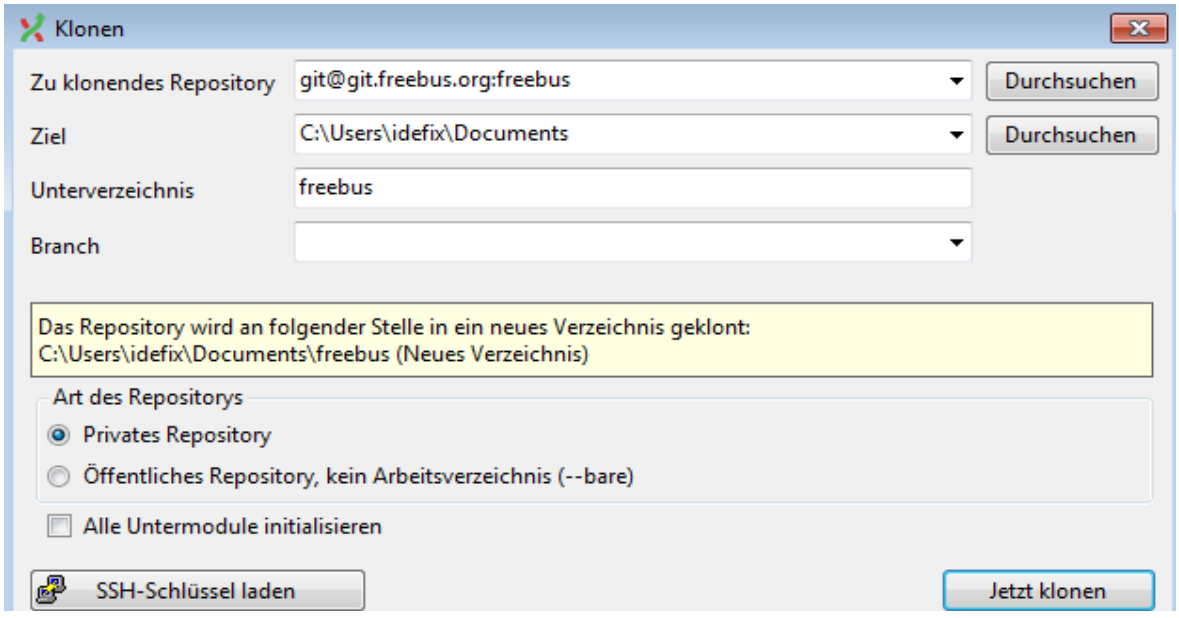
Ich selber starte den Key Agent bei mir mit dem Windows Start. Hierfür einfach einen Link im Windowsmenu Autostart anlegen und als Parameter den Privaten Key angeben, könnte dann so aussehen:

"C:\Program Files (x86)\GitExtensions\PuTTY\pageant.exe"
"C:\Users\idefix\Documents\idefix.ppk"

Der Keyagent liegt also im Verzeichnis "C:\Program Files (x86)\GitExtensions\PuTTY“ und der Key im Verzeichnis "C:\Users\idefix\Documents“.

4.3 Repository clonen mit Schreibzugriff

Git Extension starten und wieder auf Repository klonen klicken. Der Link zum klonenden Repository ändert sich nun etwas:



Dann auf Jetzt klonen klicken. Falls ihr eine Meldung Putty Security Alert bekommen solltet, prüft bitte ob der SSH Key der folgende ist:

RSA: 9d:f6:78:94:75:d9:96:d1:09:ff:e0:c3:c2:8f:48:e3
DSA: 54:b8:03:3b:46:48:81:4b:ed:5e:bc:50:aa:3f:b3:bc

Das stellt sicher, dass ihr euch mit dem richtigen Server verbindet. Es ist nur einmal nötig den Server Key zu speichern. Falls ihr nach einem Passwort gefragt werdet, habt ihr entweder euren Key wie unter 4.2SSH Key laden nicht gemacht oder der Key wurde noch nicht freigeschalten. Fangt nicht an, hier Passwörter durchzuprobieren, denn sonst sperrt der Server den Zugriff für euch komplett und ihr müsst ein paar Stunden warten, damit ihr wieder auf diesen zugreifen könnt!

Jetzt könnt ihr wie weiter oben beschrieben ganz normal arbeiten und Commits machen. Statt jetzt wieder die Daten per Email zu schicken, könnt ihr einfach auf Push klicken.