

## Inhaltsverzeichnis

1 Zugang zum GIT Repository.....	1
2 Git unter Windows installieren.....	2
3 Wie kann ich mithelfen.....	4
4 Schreibzugriff einrichten.....	7
4.1 SSH-Key anlegen.....	7
4.2 SSH Key laden.....	8
4.3 Repository clonen mit Schreibzugriff.....	8

## 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.git`.

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.

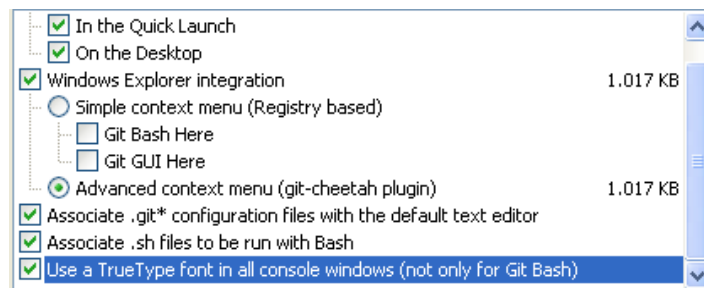
Wir wollen hier verhindern, das Projekte doppelt gemacht werden und das es leichter ist rauszufinden, wo man mithelfen kann/will.

## 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:



### ☐ Use Git Bash only

This is the most conservative choice if you are concerned about the stability of your system. Your PATH will not be modified.

### ☒ Run Git from the Windows Command Prompt

This option is considered safe and no conflicts with other tools are known. Only Git will be added to your PATH. Use this option if you want to use Git from a Cygwin Prompt (make sure to not have Cygwin's Git installed).

### ☐ Run Git and included Unix tools from the Windows Command Prompt

Both Git and its accompanying Unix tools will be added to your PATH.

**Warning: This will override Windows tools like find.exe and sort.exe. Select this option only if you understand the implications.**

### ☐ Checkout Windows-style, commit Unix-style line endings

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

### ☐ Checkout as-is, commit Unix-style line endings

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

### ☒ Checkout as-is, commit as-is

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

Danach wird Git auf dem Rechner installiert.

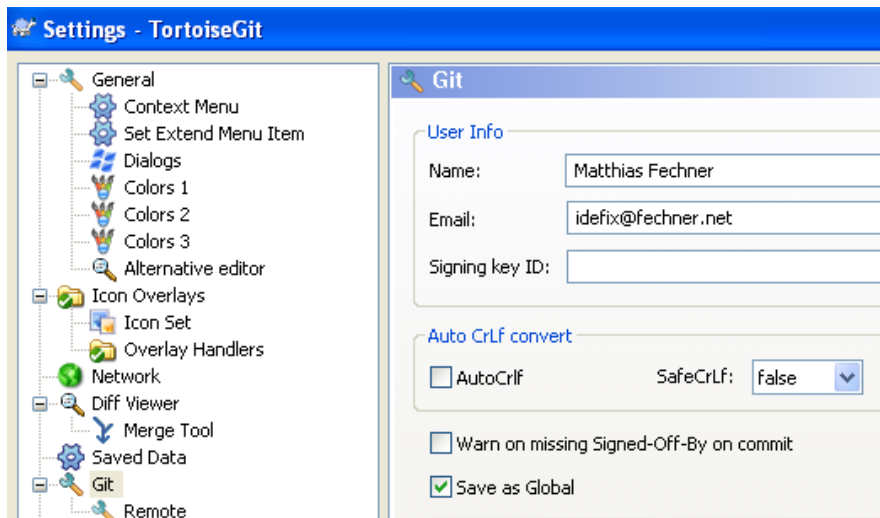
Jetzt installieren wir uns noch eine schöne GUI für Git. Ich selber benutze unter Windows ganz gerne TortoiseGit, welches hier zu finden ist:

<http://code.google.com/p/tortoisegit/wiki/Download>

Wählt die Version (32-bit oder 64-bit) aus, die zu eurem Betriebssystem passt.

Bei der Installation folgt den Standardeinstellungen.

Jetzt müssen wir noch ein paar Einstellungen an Git machen. Hierfür klickt bitte mit einem Rechtsklick in ein Verzeichnis und wählt dort TortoiseGit → Settings aus. Wählt nun Git aus und tragt jetzt euren Namen und eure Emailadresse ein:



Dann mit Ok den Dialog schliessen.

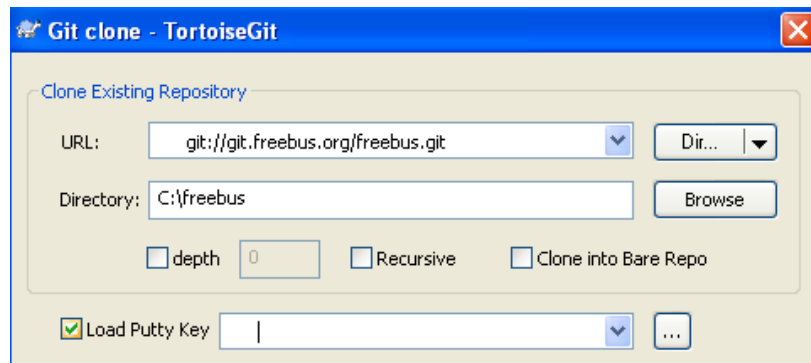
Jetzt die Git Bash starten und das folgende reinkopieren:

```
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 beizusteuern, ich zeige es hier an einem kleinen Beispiel, wie es geht.

## 3 Wie kann ich mithelfen

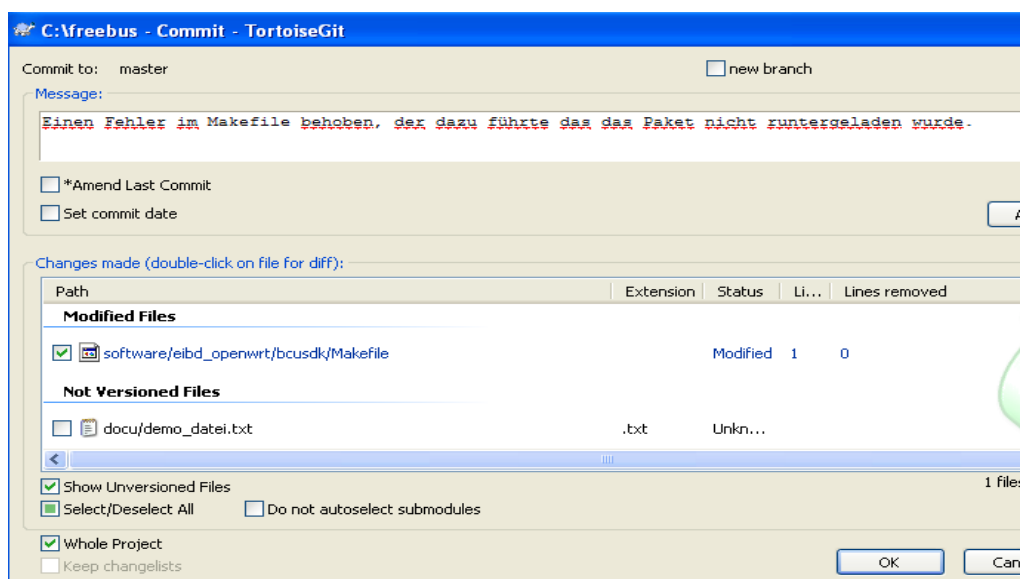


Nachdem jetzt Git installiert ist, holen wir uns erst mal eines der Repositories. Dafür einfach in einem Verzeichnis in dem ihr das Repository haben wollt einen Rechtsklick machen und Git Clone auswählen. Bei URL tragt ihr die ein, die ihr im Web unter dem Repository findet, an dem ihr etwas beisteuern 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 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.

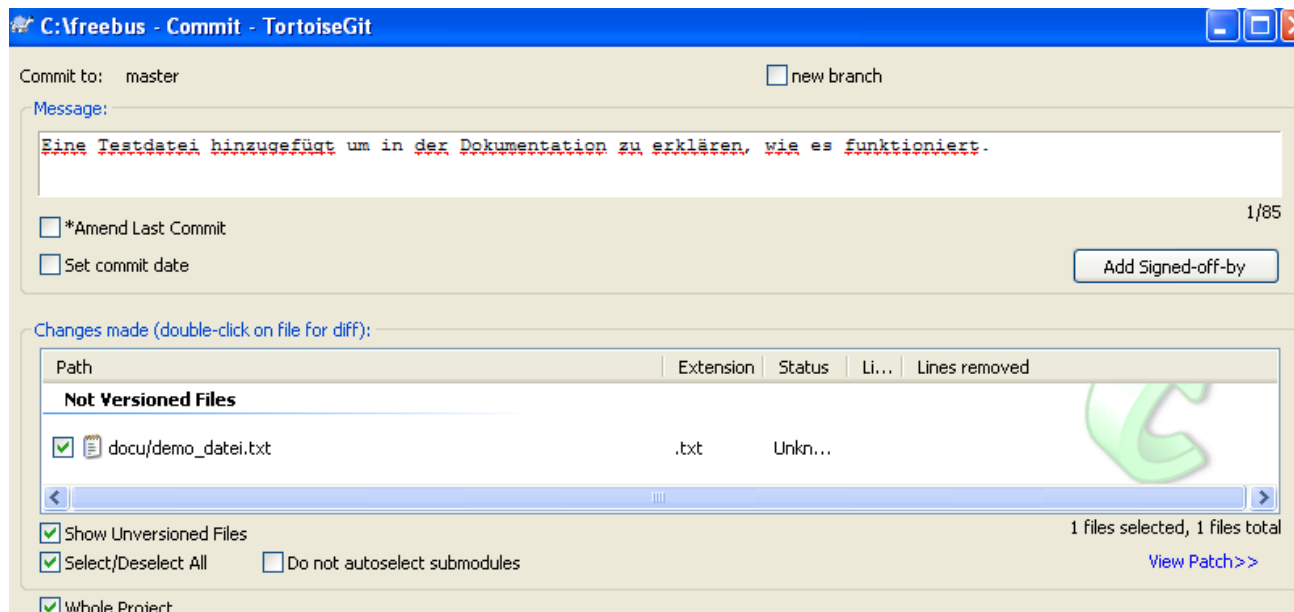
Um die Änderungen jetzt zu commiten, einfach ein Rechtsklick in das Verzeichnis machen und git Commit → Master auswählen (ich werde nicht weiter auf Branches etc. eingehen, das kann alles in entsprechenden Git Anleitungen nachgelesen werden).

Falls ihr nicht alle Dateien sehen solltet, die ihr geändert habt, wählt den Punkt „Whole Project“ aus. Die Commits sollte man logisch zusammenfassen, damit wird es später leichter nachzuvollziehen, was gemacht wurde. Da ich in diesem Beispiel zwei Sachen in unterschiedlichen Bereichen geändert habe, mache ich auch zwei Commits:



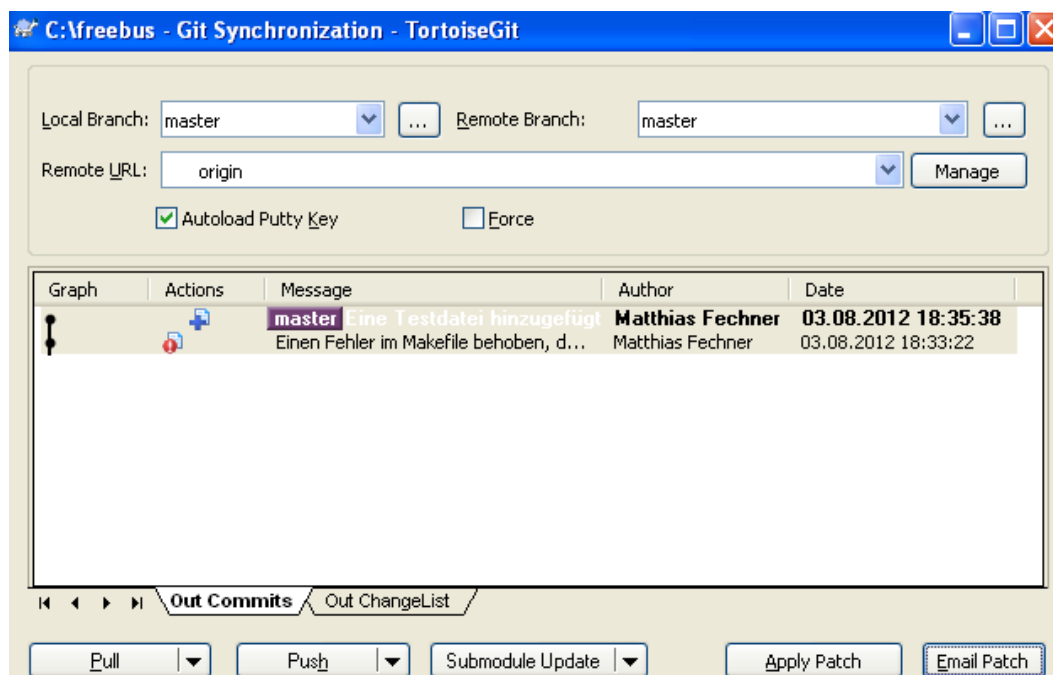
Mit Ok wird es committed, die Statusmeldung schliessen wir mit Close.

Für die zweite Änderung machen wir einen zweiten Commit:

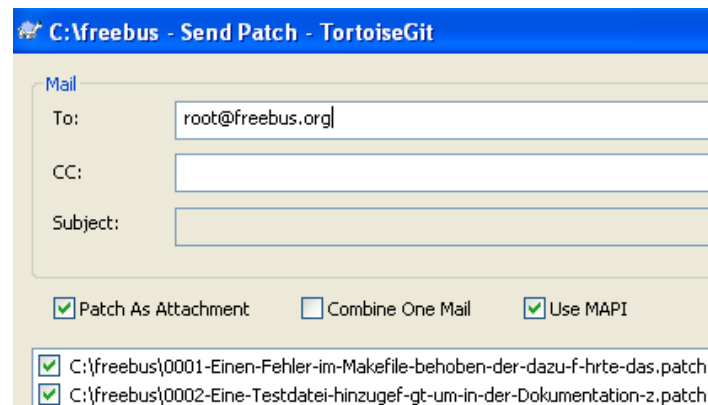


Jetzt haben wir die Änderungen lokal committed und wollen es jetzt der Community zur Verfügung stellen. Dafür erzeugen wir ein Diff, welches wir dann als Anhang in die Email Liste schicken können. Dafür einen Rechtsklick ins Verzeichnis und Git Sync auswählen.

Jetzt auswählen welche Commits wir zu Verfügung stellen wollen, einfach Klick auf den ersten Commit und Shift Klick auf den letzten. Dann rechts unten auf Email Patch klicken:



Jetzt tragen wir die Emailadresse ein (hier bitte immer die passende Liste auswählen, eine Übersicht aller Mailinglisten findet ihr auf unserer Homepage), wählen „Patch As Attachment“ und „Use MAPI“ aus und klicken ok. Jetzt sollte das Email Programm geöffnet werden und man muss nur noch aus senden drücken.



Jetzt hat euer System zwei Emails verschickt und jeder der auf der Liste steht, sieht die Mails mit dem diff an dieser Email auch. Eventuell gibt es auf der Mailingliste noch Fragen oder Verbesserungsvorschläge. Sobald alle Fragen geklärt sind, wird eure Änderung in das Repository geladen, hierbei bleibt euer Name und Emailadresse erhalten.

Nachdem euer Commit in das Repository gepusht wurde, könnt ihr einfach einen Rechtsklick in das Verzeichnis machen dort Git Sync auswählen und den Punkt Pull auswählen. Git erkennt automatisch das die Änderungen bei dir im System schon vorhanden ist und markiert diese als gepusht.

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 4 Schreibzugriff einrichten.

## 4 Schreibzugriff einrichten

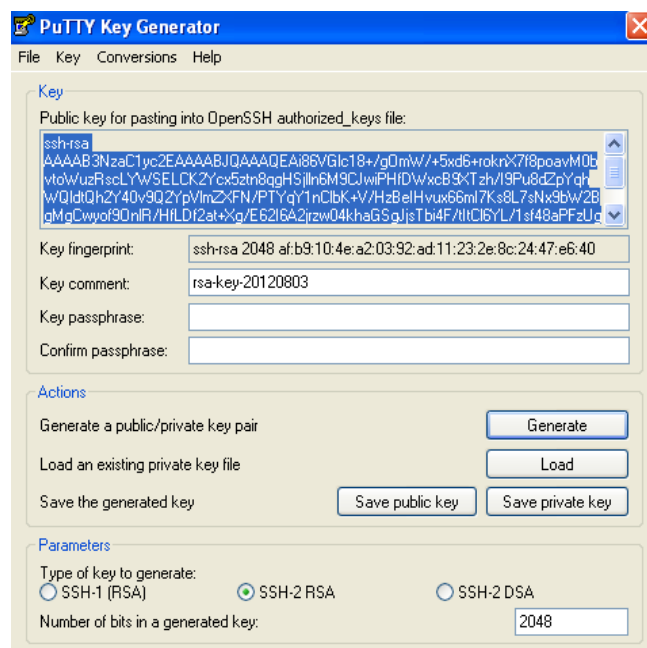
So wird der Schreibzugriff für Windows eingerichtet. Falls ihr schon einen SSH-Key habt, müsst ihr keinen neuen anlegen. Es reicht wenn ihr mir den Public Key wie in 4.1 SSH-Key anlegen beschrieben zuschickt.

### 4.1 SSH-Key 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, 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 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.

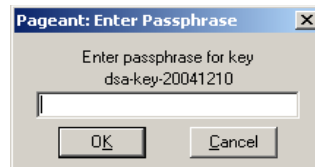
## 4.2 SSH Key laden

Jetzt brauchen wir das Programm [SSH-Key Agent](#).

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



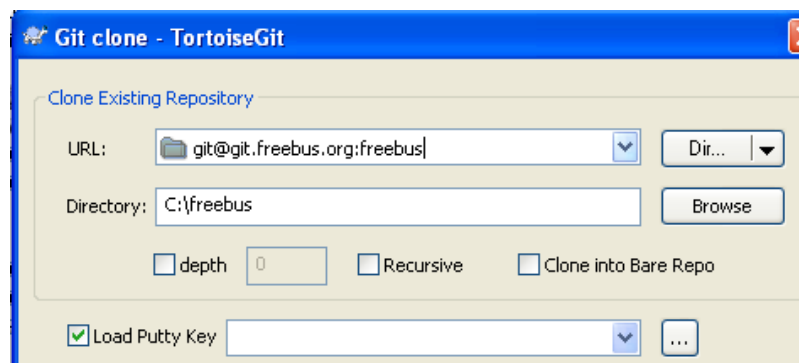
Rechtsklick drauf Add Key und jetzt den Privat Key laden den ihr vorher erstellt habt.



Jetzt die Passphrase eingeben die ihr vorher eingegeben habt.

## 4.3 Repository clonen mit Schreibzugriff

Jetzt machen wir wieder ein Clone, allerdings dieses mal über eine SSH Verbindung. Hierfür klickt ihr wieder mit einem Rechtsklick im Explorer in ein Verzeichnis und wählt Git Clone aus. Jetzt ist der Link etwas zu ändern:



Dann auf Ok 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 4.2 SSH 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.