

WinCVS

Version 1.3

Frank Grimm

fgr@micenet.de
<http://www.micenet.de>

Mario Rasser

mario@rasser-online.de
<http://www.rasser-online.de>



Voraussetzung

1. **Installation von Python (Version \geq 2.1)**
→ Download von www.python.org/download/
2. **Installation von WinCVS**
→ Download von www.cvsgui.org/download.html
3. **Installation von TCL**
→ Download von aspn.activestate.com/ASPN/Downloads/ActiveTcl

CVS Handbuch als PDF und im HTML-Format unter:

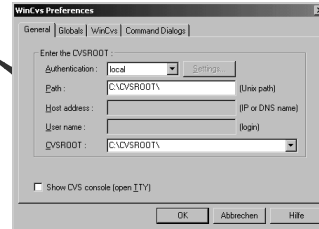
→ www.cvshome.org/docs/manual

weiter Info's unter : <http://www.cvsgui.org>

1

Erster Start:

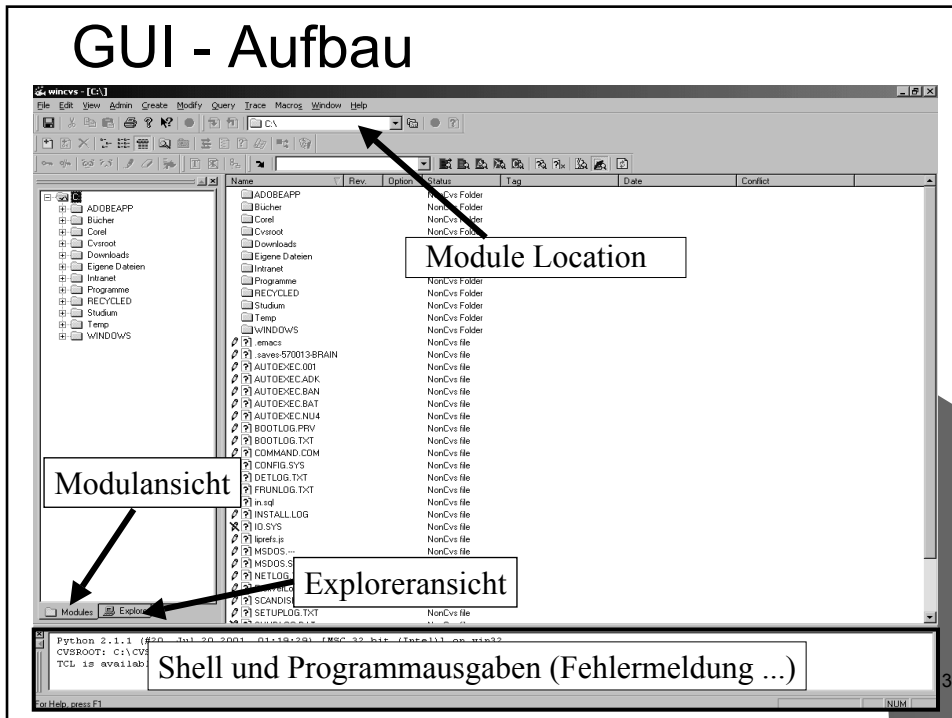
- WinCVS Preference Dialog:
 - Eingabe des CVS-Root Pfades
z.B. C:\CVSROOT
 - Authentication –Typ:
 - local
 - ssh, pserver, rhosts, ntserver ...



weiter Info's unter : <http://www.cvsgui.org>

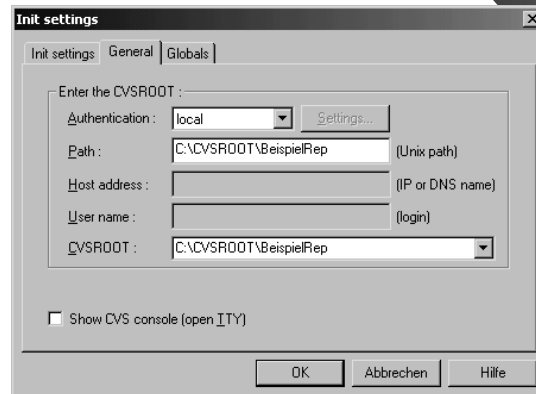
2

GUI - Aufbau



Repository erstellen:

- Menü – Create – Create a new repository ...



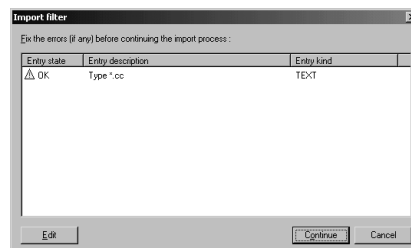
weiter Info's unter : <http://www.cvsgui.org>

4

Arbeit mit einem Repository

Module hinzufügen: (Projektbasis erstellen)

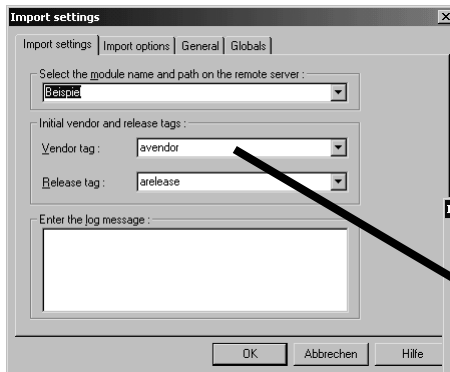
1. In Explorermodus wechseln
2. Den Ordner auswählen welcher ins CVS soll
3. Menü – Create – Import modul from selection ...



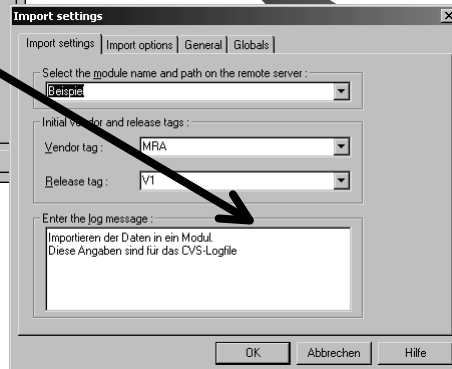
weiter Info's unter : <http://www.cvsgui.org>

5

Arbeit mit einem Repository II



Wir geben unsere Import-Einstellungen an, sowie Informationen für das CVS-Logfile



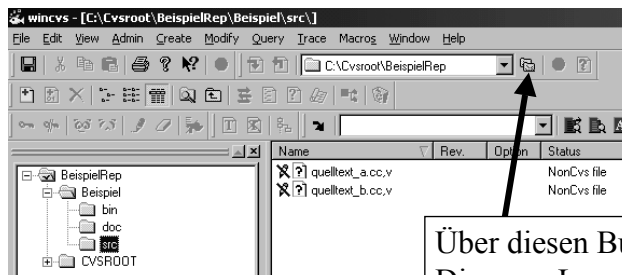
weiter Info's unter : <http://www.cvsgui.org>

6

Was ist passiert?

- CVS hat in unserer Repository-Root ein Verzeichnis angelegt (Name = Modulename)
- Alle Verzeichnisse und Dateien aus dem Projekt sind in diesem Verzeichnis enthalten

Diese Dateien benötigt CVS als Steuerdateien, sie werden nur von CVS bearbeitet und nicht vom Entwickler !



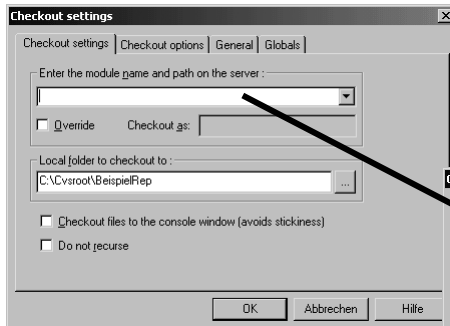
weiter Info's unter : <http://www.cvsgui.org>

Über diesen Button wurde Die neue Location ausgewählt

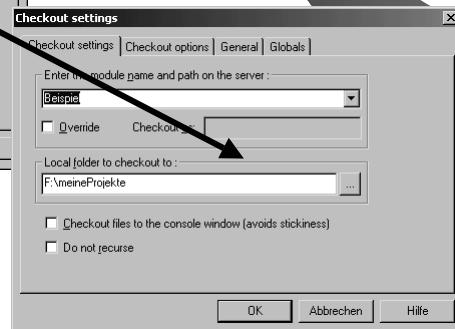
7

Wie kann ich nun mein CVS-Modul nutzen?

Module auschecken: (Arbeitsverzeichnis erstellen)



Eingabe des Modulnamen ‚Beispiel‘, sowie den Pfad an welchen das CVS-Projekt ausgecheckt werden soll.

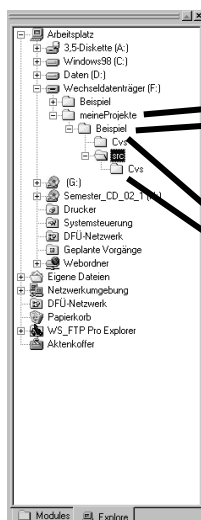


Tipp: Der Ordner in den ausgecheckt werden soll, muß noch nicht existieren.

weiter Info's unter : <http://www.cvsGui.org>

8

Wie kann ich nun mein CVS-Modul nutzen (II)?



CVS hat im Checkout-Verzeichnis (F:\meineProjekte) ein Verzeichnis (entspricht Modulname) angelegt und alle im Modul ‚Beispiel‘ vorhandenen Dateien und benötigte Verzeichnisse angelegt (höchste Version). Zusätzlich wurde in jedem Ordner ein Verzeichnis CVS angelegt, welches wiederum Steuerdaten von CVS enthält.

Übrigens: Diese CVS Verzeichnisse beinhalten Steuerdateien, welche Informationen über Pfade, Dateien und Server dieses Arbeitsverzeichnisses. Diese Dateien werden von CVS verwaltet.

In diesem ausgecheckten Modul können wir nun arbeiten, da hier unser Arbeitsordner entstanden ist

weiter Info's unter : <http://www.cvsGui.org>

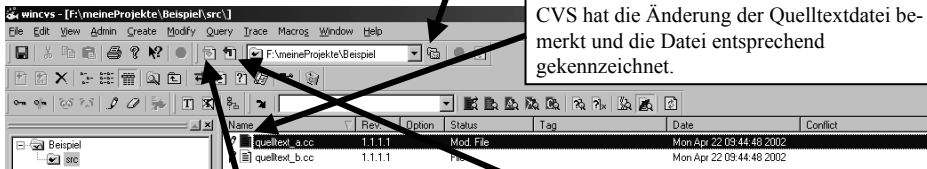
9

CVS Commit

- Quelltexte bearbeiten

Arbeitsverzeichnis für Modulsicht wählen
F:\meineProjekte\Beispiel

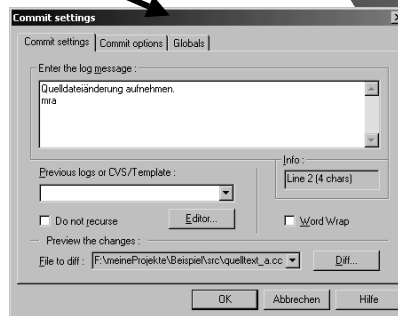
CVS hat die Änderung der Quelltextdatei bemerkt und die Datei entsprechend gekennzeichnet.



Tipp: Vor einem CVS Commit, sollte man ein CVS Update machen

Durch das CVS Commit wird die geänderte Datei in den CVS Server übernommen.

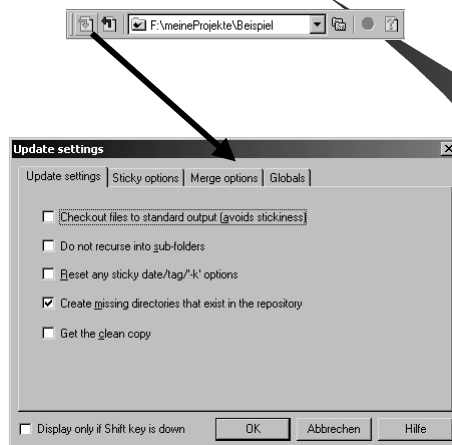
weiter Info's unter : <http://www.cvsui.org>



10

CVS Update

Änderungen anderer Entwickler in das Arbeitsverzeichnis übernehmen



weiter Info's unter : <http://www.cvsui.org>

11

Konflikte zwischen Dateien

Voraussetzung:

- Entwickler A und B haben nach einem CVS Update den gleichen Entwicklungsstand in ihrem Arbeitsverzeichnis
- Beide arbeiten unabhängig voneinander an ein und der selben Quelltextdatei z.B. quelltext_a.cc
- Frohen Mutes checkt Entwickler A seine geänderte Datei ein (CVS Commit)
- Hier entsteht noch kein Problem die Datei wird in dem CVS System aktualisiert
- Nun will Entwickler B seine Version ins CVS einchecken, würde sich CVS nicht um das Problem kümmern würde Entwickler B, die Arbeit von Entwickler A einfach überschreiben.

DIESER FALL MUSS VOM CVS ABGEFANGEN WERDEN

weiter Info's unter : <http://www.cvsgui.org>

12

CVS bringt nach dem CVS Commit Versuch von Entwickler B folgende Fehlermeldung in der Shell:

```
cvsv commit -m "meine Änderung Entw. B\n" quelltext_a.cc (in directory F:\ein_anderer_Entwickler\Beispiel\src)\ncvs95 commit: Up-to-date check failed for 'quelltext_a.cc'\ncvs95 [commit aborted]: correct above errors first |  
*****CVS exited normally with code 1*****
```

Also machen wir ein CVS Update da unsere Daten nicht aktuell sind, wie aus der CVS Fehlermeldung zu erkennen.

Nach dem CVS Update sehen wir folgende Ausgaben in der WinCVS GUI:

Name	Rev.	Option	Status	Tag	Date	Conflict
./#quelltext_a.cc.1.2			NonCvs file			
./quelltext_a.cc	1.3		Conflict		Mon Apr 22 18:52:40 2002	Result of merge
./quelltext_b.cc	1.1.1.1				Mon Apr 22 09:44:48 2002	

Es ist eine Kopie unseres Quelltextes entstanden (./#quelltext_a.1.2) und es wird der Konflikt in quelltext_a.cc angezeigt. In dieser Datei sind die Änderungen der beiden Version von Entwickler A und B enthalten.

weiter Info's unter : <http://www.cvsgui.org>

13

Konflikte lösen

Hier die Datei ‚quelltext_a.cc‘, welche im CVS mit Conflict gekennzeichnet ist. CVS hat die Stellen an dem der Konflikt auftrat gekennzeichnet.

```
0 10 20 30 40 50
1 <<<<<< quelltext_a.cc
2 //Hier steht jetzt Quelltext
3 //Hier schreibt Entwickler B seinen Quelltext*****
4 //Hier steht jetzt Quelltext
5 //Hier hat Entwickler A was gemacht>>>>>>> 1.3
6
```

Datei ‚quelltext_a.cc‘ in einem Texteditor

Nun kann Entwickler B die Änderungen vergleichen und die Datei konsistent machen. So das dem Projekt keine Daten verloren gehen.

Nachdem dies geschehen ist kann die Datei eingecheckt werden (CVS Commit) und der **Versionskonflikt wurde gelöst.**

Name	Rev.	Option	Status	Tag	Date	Conflict
? #quelltext_a.cc.1.2			NonCvs file			
? quelltext_a.cc	1.4		File		Mon Apr 22 19:05:42 2002	
? quelltext_b.cc	1.1.1.1		File		Mon Apr 22 09:44:48 2002	

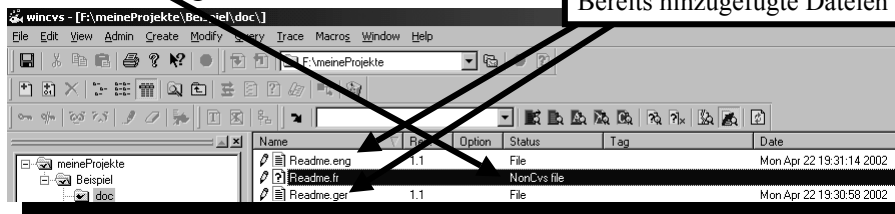
weiter Info's unter : <http://www.cvsgui.org>

14

CVS Add

Hinzufügen noch nicht vorhandener Dateien ins CVS:

- Erstellen der Dateien durch entsprechenden Editor
- WinCVS in Modulmodus wechseln
- Ordner markieren und auf ‚Add Selection‘ anklicken
- neue Datei markieren und ebenfalls durch Add Selection hinzufügen



- Jetzt müssen via Commit die Dateien noch eingecheckt werden

weiter Info's unter : <http://www.cvsgui.org>

15

Zusammenfassung

- Repository erstellen** - nur beim ersten mal
- Modul importieren** - für jedes Projekt im Repository
- CVS Checkout** - zum erstellen des Arbeitsverzeichnisses (muß jeder Entwickler machen)
- CVS Update** - Aktualisierung des Arbeitsverzeichnisses. Änderungen anderer Entwickler übernehmen
- CVS Commit** - Änderungen ins CVS übernehmen und somit bereitstellen für andere Entwickler
- CVS Add** - Hinzufügen noch nicht vorhandener Dateien ins CVS

weiter Info's unter : <http://www.cvsgui.org>

16

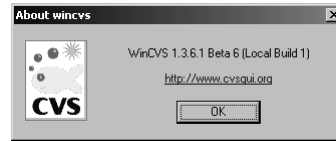
Weiter Features:

- Auschecken einer bestimmten Release-Version
 - Nutzung vorhandener CVS Repositories via Internet
 - Erstellen von Tag's
 - Löschen von Modulen und Dateien
 - Rückwärtsgehen zu älteren Versionen
 - Kontrollen durch Logfiles
 - Statusinformationen zu Modulen und Dateien
 - Erstellen von Branche's
- u.v.m.

weiter Info's unter : <http://www.cvsgui.org>

17

Quellen und Links:



- <http://www.cvsgui.org>
- <http://www.cvshome.org>
- CVS-Cookbook

- WinCVS Version 1.3 – Users Guide von Don Harper

- Diese PowerPoint Präsentation (gezippt):
<http://www.rasser-online.de/projekte/cvs/vortrag.zip>

- PDF-Dateien zu diesem Vortrag:
http://www.rasser-online.de/projekte/cvs/vortrag_sw.pdf
http://www.rasser-online.de/projekte/cvs/vortrag_color.pdf

weiter Info's unter : <http://www.cvsgui.org>

18