

Fachhochschule Bochum

Fachbereich Wirtschaft

<http://www.fh-bochum.de/fb6/>

EDV-Labor

Prof. Dr. Bernd Blümel

## Inhaltsverzeichnis

1. Begriffserklärungen: .....	3
2. Installation: .....	4
3. Voreinstellung von CVSROOT unter Suse Linux .....	5
4. Checkout .....	5
5. Commit .....	6
6. Status .....	6
7. Aktualisierung (update) .....	7
8. Diff .....	7
9. Konflikt .....	9
10. Log .....	9
11. Änderungen rückgängig machen .....	10
12. taggen (Momentaufnahme erstellen) .....	11
13. Datei hinzufügen .....	12
14. Datei löschen .....	12
15. Verzeichnis löschen .....	13
16. Literatur: .....	13

## 1. Begriffserklärungen:

### **Revision**

Eine Veränderung an einer Datei oder Menge von Dateien, die durch einen Commit abgeschlossen wurde. Eine Revision ist eine Momentaufnahme eines sich ständig verändernden Projektes.

### **Archiv**

Die Hauptkopie, in der CVS die vollständige Revisionshistorie eines Projektes speichert. Jedes Projekt hat genau ein Archiv.

### **Arbeitskopie**

Die Kopie, mit der gearbeitet wird und die tatsächlich verändert wird. Es kann mehrere Arbeitskopien eines bestimmten Projektes geben; im Allgemeinen hat jeder Entwickler seine eigene Kopie.

### **Checkout**

Eine Arbeitskopie von dem Archiv anfordern. Die angeforderte Kopie stellt den Zustand des Projektes zu dem Zeitpunkt dar, in welchem die Kopie angefordert wurde. Wenn Sie oder andere Entwickler Veränderungen vornehmen, müssen commit und update durchgeführt werden, um die eigenen Veränderungen zu »veröffentlichen« und die der anderen Entwickler sehen zu können.

### **Commit**

Senden der eigenen Veränderungen zum zentralen Archiv. Auch Check-in genannt.

### **Log-Nachricht**

Ein Kommentar, der bei einem Commit einer Revision angehängt wird und die vorgenommenen Veränderungen beschreibt. Andere Entwickler können durch die Log-Nachrichten blättern und erhalten so die Antwort auf die Frage, was in dem Projekt passiert ist.

### **Add**

Hinzufügen einer Datei, die bis dato noch nicht im Repository erfasst ist.  
(Das Hinzufügen geschieht in zwei Schritten) erst add, dann commit.

## Remove

Löschen einer Datei aus dem Repository. Das Löschen erfolgt in mehreren Schritten. Erst wird die Datei physikalisch aus der Arbeitskopie gelöscht.

Anschließend aus dem Repository "removed" und dann die "virtuelle gelöschte Datei von dem Arbeitsverzeichnis in das Repository "committed".

Das mag sich jetzt ein bisschen verwirrend anhören, jedoch ist in diesem Tutorial ein Beispiel das dies (hoffentlich :-) einleuchtend erklärt.

## 2. Installation:

Das CVS ist bei Suse standardmäßig vorinstalliert.

Ein Repository (Archiv unter CVS) wird wie folgt eingerichtet:

Erstellen des Verzeichnisses in welchem das CVS seine Daten speichern soll:

```
cm@helga:~> mkdir /home/cvs
```

Anschließend wird folgender Befehl eingegeben:

```
cm@helga:~> cvs -d /home/cvs init
```

damit erzeugt man das Verzeichnis /home/cvs/CVSROOT

In diesem werden dann die Informationen für die Verwaltung abgespeichert.

Legt man nun ein Repository an, so ist dies physikalisch unter /home/cvs/intranet.

Das Verzeichnis /home/cvs/intranet wird beim Import automatisch von CVS angelegt. Sollte man den init - Befehl anwenden, obwohl das CVSROOT schon erzeugt wurde, passiert nichts. Den Schalter -d /home/cvs könnte man sich sparen wenn man folgendes an der Shell eingibt:

```
cm@helga: /> CVSROOT=/home/cvs  
cm@helga: /> export CVSROOT
```

jetzt testen wir gleich ob die Variable gesetzt wurde:

```
cm@helga: /> echo $CVSROOT  
/home/cvs  
cm@helga: />
```

Jetzt wechselt man in das Verzeichnis, welches man unter CVS stellen möchte

```
cm@helga:~> cd /srv/www/htdocs/intranet
```

Anschließend wird das Verzeichnis ins CVSROOT importiert (neu angelegt)

```
cm@helga:> cvs -d /home/cvs import -m "Erster Import"  
intranet cm V20021015
```

- -d /home/cvs gibt das Verzeichnis an, in welchem CVSROOT ist
- import erzeugt ein Repository

- m "Erster Import" ist die Log – Nachricht die man anschließend im Repository sieht
- intranet gibt den Namen des Repository an
- cm ist der Hersteller-Tag, hier kann man schreiben was man möchte, hat zwar eine Bedeutung, aber nur wenn man ein CVS-Guru ist :-)
- V20021015 ist die Versions Marke. Auch hier ist Text frei wählbar
- ich habe mich fürs aktuelle Datum entschieden

Wenn nach Absetzen des Befehls folgendes über den Bildschirm flimmert, hat es funktioniert:

```
I intranet/CVS
N intranet/config.inc.php
N intranet/faq.php
N intranet/index.html
N intranet/visitor.php
N intranet/index.php
N intranet/index_original.php
N intranet/news.php
N intranet/help.html
```

Es werden die Dateien aufgelistet, welche importiert werden.

Mit dem Kommando import hat man jedoch noch keine arbeitsfähige Kopie. Diese erhält man erst durch das auschecken.

### 3. Voreinstellung von CVSROOT unter Suse Linux

Öffnen der Datei `.bashrc` im home-directory (die Datei ist versteckt und kann mit Ansicht-> versteckte Dateien anzeigen im Konqueror sichtbar gemacht werden) und Einfügen der Zeilen:

```
# set cvs root
export CVSROOT=/home/cvs
```

### 4. Checkout

```
cm@helga:~ > mv intranet save_intranet
cm@helga:~ > cvs -d /home/cvs checkout intranet
```

**Beachten:** Die Kopie wird bei o.g. Befehl in das Verzeichnis kopiert, in welchem man sich gerade befindet.

```
cm@helga:~ > ls
intranet  save_intranet
```

```
cm@helga:~ >
```

hat funktioniert, eine Arbeitskopie wurde ausgecheckt.

## 5. Commit

```
cm@helga:~ > cd intranet
cm@helga:~/intranet >
cm@helga:~/intranet > cvs -d /home/cvs commit -m "Aenderung
an der Dokumentation des Intranet"
Checking in index.php;
/home/cvs/intranet/index.php,v <-- index.php
new revision: 1.2; previous revision: 1.1
done
cm@helga:~/intranet >
```

Hier wird wieder der Schalter -m benutzt. Benutzt man diesen nicht, so öffnet sich unter Linux der Editor vi, und CVS erwartet eine Beschreibung.

Schließen kann man den vi mit :q oder alternativ die brute force Methode: :qa!

Sollte der Fall eintreten, dass zwei Entwickler an ein und derselben Datei arbeiten und genau die gleiche Stelle verändern, so bricht der commit mit einer Meldung ab.

CVS weiß nicht, welche Änderung es nun ins Repository übernehmen soll und welche nicht.

Hier müssen sich dann die Entwickler besprechen. In den betroffenen Dateien wird eine Markierung durch das CVS angelegt, d.h. die betreffenden Abschnitte sind gekennzeichnet.

### Achtung:

Vor jedem Commit muss ein Update durchgeführt werden (s. Punkt Update weiter unten).

## 6. Status

mit dem Kommando Status kann man einen Versionsvergleich durchführen und bekommt diesen angezeigt.

```
cm@helga:~/intranet > cvs -d /home/cvs status index.php
=====
File: index.php          Status: Up-to-date
  Working revision:      1.2      Wed Oct  9 15:32:26 2002
  Repository revision:  1.2
/home/cvs/intranet/index.php,v
  Sticky Tag:            (none)
  Sticky Date:           (none)
  Sticky Options:        (none)

cm@helga:~/intranet >
```

## 7. Aktualisierung (update)

Veränderungen von anderen Entwicklern vom Archiv in die eigene Arbeitskopie aufnehmen und aufzeigen, ob die eigene Arbeitskopie noch nicht durch commit abgeschickte Veränderungen enthält.

Enthält die Arbeitskopie eine Datei, die noch nicht im Repository vorhanden ist, so bekommt man dies durch ein Fragezeichen vor dem Dateinamen angezeigt:

```
? classes/Company.inc.php
? classes/StudyCourse.inc.php
M displayFunctions/displayHelperFunctions.inc.php
```

Das Fragezeichen vor dem Dateinamen besagt also, dass diese Datei nicht im CVS vorhanden ist. Sie muss mit add hinzugefügt werden und dann committed.

```
cm@helga:~ > cvs -d /home/cvs update intranet
cvs update: Updating intranet
M intranet/index.php
cvs update: Updating intranet/Marks
cvs update: Updating intranet/Marks/classes
cvs update: Updating intranet/Marks/classesSicher
cvs update: Updating intranet/Marks/images
cvs update: Updating intranet/Marks/javascripts
```

Das M steht für Merge, d.h. hier werden zwei unterschiedliche Dateien zusammengeführt.

Möchte man nur eine einzelne Datei updaten:

```
cm@helga:~ > cvs -d /home/cvs update index.php
```

Der Unterschied zwischen Update und Diff besteht darin, dass Update die Veränderungen in das Arbeitskopie einbringt und Diff die Veränderungen anzeigt. Wenn also Entwickler A eine Datei verändert hat, und Entwickler B ein Update macht, wird die Veränderung von Entwickler A in die Arbeitskopie von Entwickler B eingearbeitet. Möchte Entwickler B zuvor die Veränderungen sehen, benutzt er das Diff Kommando.

## 8. Diff

```
cm@helga:~ > cvs -d /home/cvs diff
cvs diff: Diffing intranet
Index: intranet/index.php
=====
RCS file: /home/cvs/intranet/index.php,v
retrieving revision 1.1.1.1
diff -r1.1.1.1 index.php
13c13,14
```

```
<
---
> //Jetzt wird CVS getestet, nachfolgend wird die Variable
$ROOT per echo ausgegeben
> echo $ROOT;
```

man sieht in welcher Datei (index.php) eine Änderung vorgenommen wurde.

Zeile 13 Beginn der Änderung

c bedeutet: die Kopie wurde geändert, würde a hier stehen wäre das Archiv geändert worden

Zeilen 13 und 14 wurden geändert

```
cm@helga:~ > cvs -d /home/cvs -Q diff
Index: intranet/index.php
=====
RCS file: /home/cvs/intranet/index.php,v
retrieving revision 1.1.1.1
diff -r1.1.1.1 index.php
4c4
<
---
> //TestKommentar zum CVS-Testen
cm@helga:~ >
```

mit der Option -Q weist man CVS an, nicht anzuzeigen, wo es gerade arbeitet, schon ein bisschen übersichtlicher das Ganze

```
cm@helga:~ > cvs -d /home/cvs diff -c
cvs diff: Diffing intranet
Index: intranet/index.php
=====
RCS file: /home/cvs/intranet/index.php,v
retrieving revision 1.1.1.1
diff -c -r1.1.1.1 index.php
*** intranet/index.php 2002/10/09 02:47:03 1.1.1.1
--- intranet/index.php 2002/10/09 15:32:26
*****
*** 10,16 ****

    require_once (" $ROOT_DIR/commonHtml/header.html");

!

?>
```

```
--- 10,17 ----  
  
    require_once ("$ROOT_DIR/commonHtml/header.html");  
  
    ! //Jetzt wird CVS getestet, nachfolgend wird die Variable  
    $ROOT per echo ausgegeben  
    ! echo $ROOT;  
  
    ?>  
cm@helga:~ >
```

oben wird die Archivdatei angezeigt, drunter die der Arbeitskopie.

Man sieht hier, dass 2 Zeilen eingefügt wurden. Das Ausrufezeichen zeigt an, dass die Zeile zwischen der neuen und alten Datei unterschiedlich ist.

Auch hier kann man das Diff von nur einer Datei anzeigen lassen:

```
cm@helga:~ > cvs -d /home/cvs diff
```

## 9. Konflikt

Situation, in der zwei Entwickler Veränderungen im gleichen Teil der gleichen Datei per commit abzuschicken versuchen. CVS bemerkt solche Konflikte und benachrichtigt die Entwickler, aber die Entwickler müssen diese selbst auflösen.

## 10. Log

Veränderungen lesen

```
cm@helga:~/intranet > cvs -d /home/cvs log index.php  
  
RCS file: /home/cvs/intranet/index.php,v  
Working file: index.php  
head: 1.2  
branch:  
locks: strict  
access list:  
symbolic names:  
    Start: 1.1.1.1  
    cm: 1.1.1  
keyword substitution: kv  
total revisions: 3;      selected revisions: 3  
description:  
-----  
revision 1.2  
date: 2002/10/09 15:48:39;  author: cm;  state: Exp;  lines:  
+2 -1
```

```
Aenderung an index.php
-----
revision 1.1
date: 2002/10/09 02:47:03;  author: bb;  state: Exp;
branches: 1.1.1;
Initial revision
-----
revision 1.1.1.1
date: 2002/10/09 02:47:03;  author: bb;  state: Exp;  lines:
+0 -0
Erster Import Intranet -> CVS
=====
cm@helga:~/intranet >
```

## 11. Änderungen rückgängig machen

Eine Änderung rückgängig machen bedeutet, eine vorherige Version wieder aktuell zu machen. Angenommen, aktuell ist Version 1.4 und die Version die man wieder haben möchte ist 1.3, so ist das Ergebnis der Aktion nicht eine Version 1.3 sondern eine Version 1.5. Verfährt man so, ist das Rückgängigmachen ein Teil der History und jederzeit einsehbar. Möchte man Änderungen rückgängig machen, so gibt es zwei verschiedene Methoden: eine mühevollere und eine schnelle.

Hier wird die schnelle Methode beschrieben:

Folgende Situation:

Entwickler A hat die Datei index.php geändert und committed. Diese Datei hat jetzt die Version 1.4. Entwickler B hat auch eine index.php, aber in der Version 1.3.

Entwickler B schaut nach, was Entwickler A in Datei index.php geändert hat und entscheidet sich für eine Rückgängigmachung. Es sind dazu folgende Arbeitsschritte von Entwickler B notwendig:

- cvs update:  
damit holt sich Entwickler B die Version 1.4 der Datei index.php aus dem Archiv
- Überspielen der Version 1.4. mit der Version 1.3
- committen der Datei index.php, so dass anschließend die Version 1.5 die aktuelle ist (incl. Rückgängigmachung der Version 1.4).

Das Kommando für das Reaktivieren von alten Versionen lautet z.B.:

```
cm@helga:/home/cm/intranet> cvs update -j 1.2 -j 1.3
index.php
```

Gibt es bei einer Reaktivierung unüberbrückbare Differenzen, so gibt CVS eine Meldung aus. Anschließend kann man im Source der Datei die Stellen markiert vorfinden:

```
<<<<<< index.php
=====
//Jetzt wird CVS getestet, nachfolgend wird die Variable
$ROOT per echo ausgegeben
echo $ROOT;
echo "Test";
>>>>>> 1.3
?>
```

CVS verwaltet Binäre Dateien nicht sauber. Dies liegt zum einen am unterschiedlichen Abschluss der Zeilenenden (Windows erwartet ein LF und CR (Line-Feed und Carrige Return), Unix erwartet nur ein LF; zum anderen sucht CVS in den Dateien nach speziellen Zeichenketten, auch RCS-Schlüsselwörter genannt. Findet CVS so eine spezielle Zeichenkette, so wird diese durch z.B. Revisionsinformationen und andere nützliche Dinge ersetzt.

Enthält eine Datei z.B.:

```
$Revision$
```

so ersetzt CVS z.B. bei jedem commit dies durch:

```
$Revision 1.3$
```

Was bei Textdateien sehr nützlich ist, ist hingegen bei z.B. Bilddateien fatal. CVS durchforstet eine Bilddatei und sobald es ein Schlüsselwort findet, ersetzt dies CVS durch eigene Informationen. Was hierbei als Ergebnis präsentiert wird, kann sich jeder denken. Es besteht die Option, mit dem Schalter `-ko filename` die Schlüsselworterkennung und die Zeilenendumwandlung auszuschalten.

```
cm@helga:/home/cm/intranet> cvs -d /home/cvs add -kb home.jpg
```

möchte man nur die Schlüsselworterkennung abschalten, so lautet das Kommando:

```
cm@helga:/home/cm/intranet> cvs -d /home/cvs add -ko home.jpg
```

## 12. taggen (Momentaufnahme erstellen)

Es gibt zwei Möglichkeiten, Momentaufnahmen zu erstellen:

1. Zeitstempel
2. Marken

Die Methode mit dem Datum wird im Handbuch behandelt. Hier wird die Methode mit den Marken erläutert.

Taggen unterscheidet sich vom normalen commit, dass hier keine Dateien geändert werden, sondern die Dateien im Archiv „auf einen Stand“ gebracht werden hinsichtlich Release. Es kann durchaus sein, dass die Datei index.php die Versionsnummer 1.1 hat und die Datei foo.php die Version 1.5. Dennoch gehören beide Dateien zum Release „First stable“.

```
cm@helga:/.../intranet > cvs -d /home/cvs/ -q tag  
FinalRelease
```

Dieser Befehl fixiert eine Momentaufnahme. Diese kann später jederzeit wieder rekonstruiert werden in einem Arbeitsarchiv.

Möchte man genau diese Version wieder herstellen, so benutzt man folgendes Kommando:

```
cm@helga:/.../intranet > cvs -d /home/cvs/ checkout -r  
FinalRelease intranet
```

### 13. Datei hinzufügen

Das Hinzufügen einer Datei zu einem Repository geschieht in zwei Schritten. Erst wird ein add-Befehl ausgeführt und dann wird die Datei committed.

```
cm@ushiro:/home/intranet > cvs add EineNeueDatei.php  
  
cvs add: scheduling file ` EineNeueDatei.php ' for addition  
cvs add: use 'cvs commit' to add this file permanently  
  
cm@ushiro:/home/intranet > cvs commit -m "Ab ins Repository"  
EineNeueDatei.php  
RCS file: /home/intranet/classes/EineNeueDatei.php ,v  
  
done  
  
Checking in EineNeueDatei.php;  
  
/home/cvs/intranet/classes/EineNeueDatei.php ,v <--  
EineNeueDatei.php  
initial revision: 1.1  
done
```

### 14. Datei löschen

Um eine Datei zu löschen sind drei Arbeitsschritte notwendig:

1. Zunächst wird die Datei physikalisch aus der Arbeitskopie gelöscht:

```
cm@helga:~/intranet > rm wegMitDerDatei.php
```

2. Dann wird dem CVS mitgeteilt das es die Datei aus dem Repository löschen soll:

```
cm@helga:~/intranet > cvs -d /home/cvs/ remove  
wegMitDerDatei.php
```

```
cvs remove: scheduling `wegMitDerDatei.php' for removal  
cvs remove: use 'cvs commit' to remove this file permanently
```

3. Anschließend wird die nicht mehr in der Arbeitskopie vorhandene Datei in das Repository committed:

```
cm@helga:~/intranet > cvs -d /home/cvs commit -m "Diese  
Datei ist sowas von gelöscht" wegMitDerDatei.php  
Removing wegMitDerDatei.php;  
/home/cvs/intranet/wegMitDerDatei.php ,v <--  
wegMitDerDatei.php  
new revision: delete; previous revision: 1.1  
done
```

## 15. Verzeichnis löschen

Das CVS überwacht leider nicht die Verzeichnisse. Deswegen gibt es keinen offiziellen Befehl um ein Verzeichnis aus dem Repository zu löschen.

Daher wird das Verzeichnis auf physikalischen Weg durch `rm -R Verzeichnisname` gelöscht.

## 16. Literatur:

<http://www.cvshome.org>