

NT-Seminarvortrag am 11. Dezember 2001

**Einführung in die
Versionsverwaltung CVS**

von Gordon Cichon



Inhalt

- ▶ Einführung und Begriffserklärung
- ▶ CVS im Alltag
 - „Auschecken“ eines Projekts
 - Anzeige von Änderungen
 - „Einchecken“ von Änderungen
 - Umgang mit Konflikten
 - Dateien hinzufügen und löschen
- ▶ CVS für Fortgeschrittene
 - CVS Schlüsselwörter und Binär-Dateien
 - Anlegen von „Releases“
- ▶ Weitere Informationsquellen



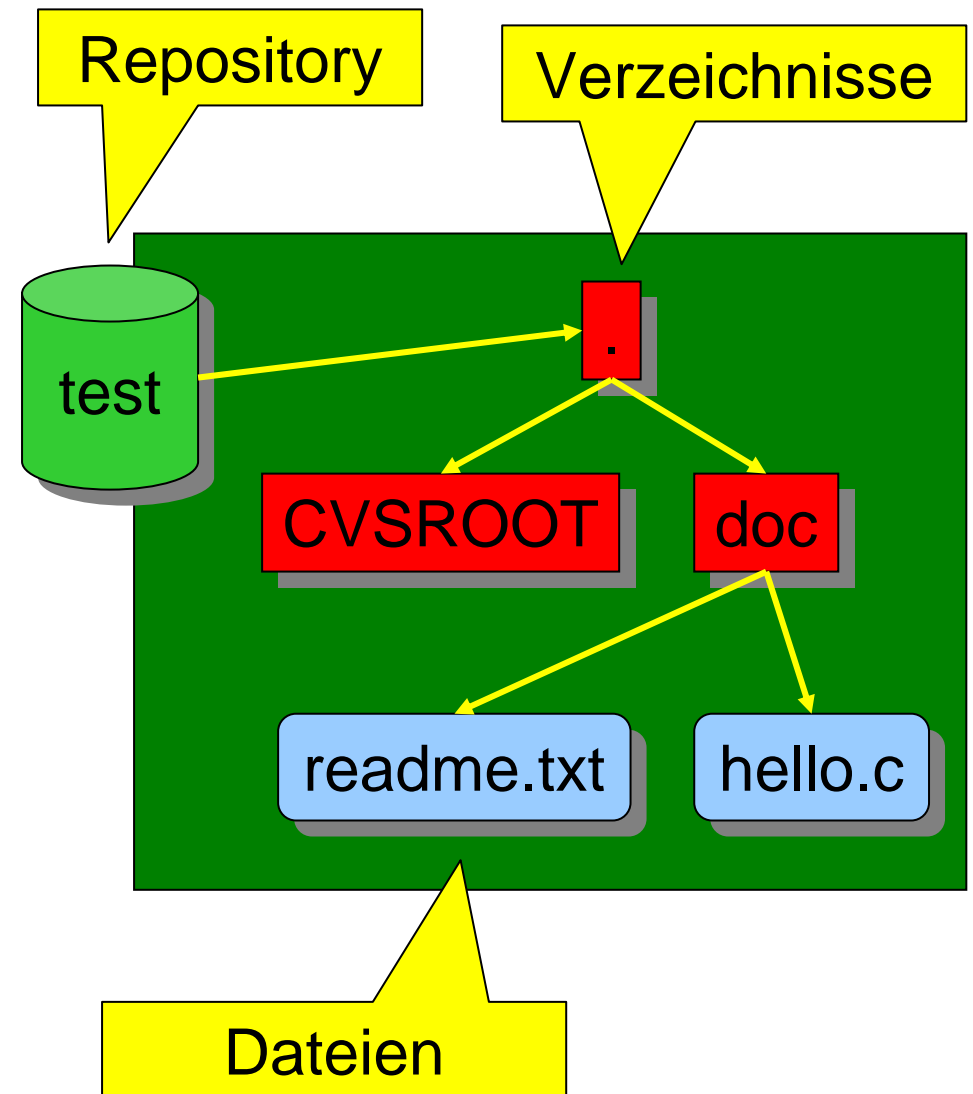
Einführung

- ▶ CVS = „Concurrent Versioning System“
- ▶ Verwaltet und archiviert Änderungen an Dateien
- ▶ Was kann CVS gut:
 - Abspeicherung von früheren Versionen von Dateien
 - Gruppenarbeit auf einem Satz Dateien
 - Abspeichern eines Verzeichnisbaumes
 - Effektives Zusammenführen von gleichzeitigen Änderungen an Text-Dateien
- ▶ Wo kann CVS noch verbessert werden:
 - Kein Sperren (*locking*) von Dateien
 - Kein „Build-System“, d.h. CVS ersetzt nicht *make*



Begriffserklärung

- ▶ **Repository oder Datenbank**
Unabhängiges Satz von Dateien, die von CVS verwaltet werden
- ▶ **Modul** = Verzeichnis
- ▶ **Checkout** = Anfertigen einer lokalen Kopie der Dateien des Repository
- ▶ **Commit = Checkin** = Übertragen von Änderungen aus lokaler Kopie ins Repository
- ▶ **Diff** = Änderungen zwischen verschiedenen Versionen einer Datei
- ▶ **Merge** = Zusammenführen von Änderungen verschiedener Versionen
- ▶ **Conflict** = Scheitern des Zusammenführens



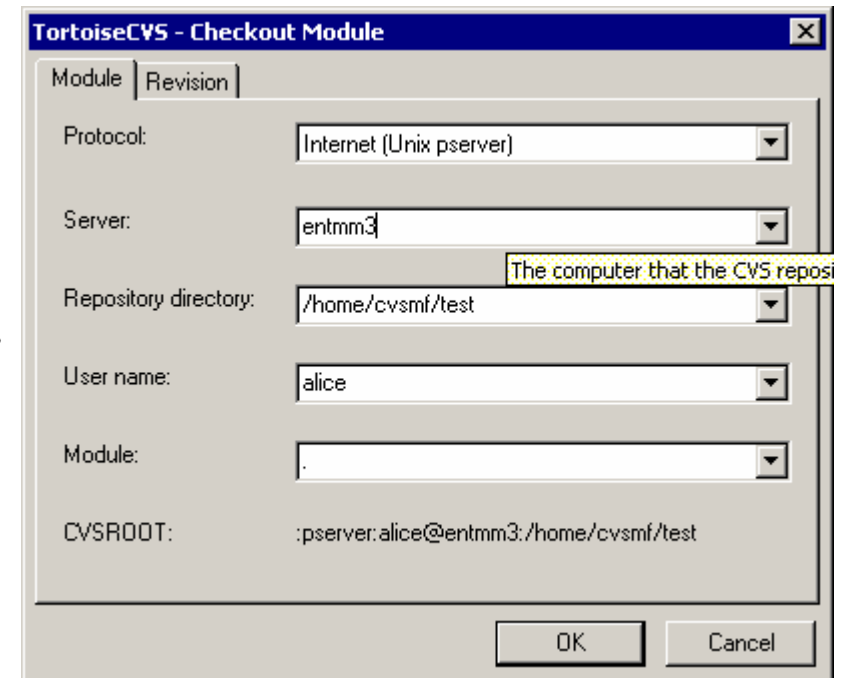
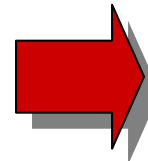
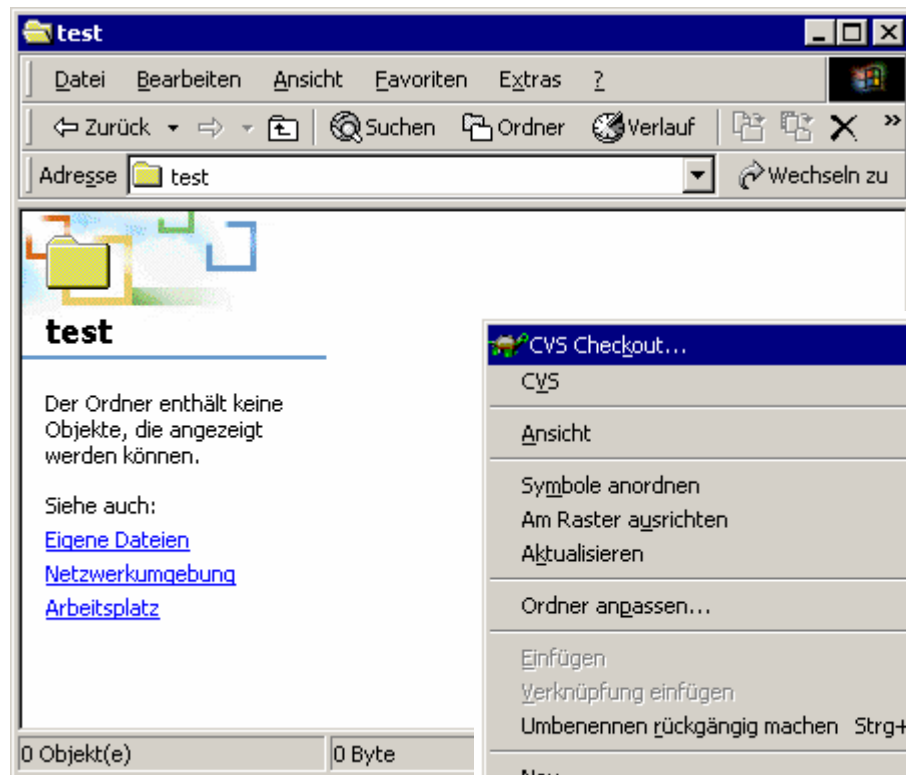
Erste Schritte: Auschecken

Benutzername

Server

- ▶ Voraussetzung: CVS-Repository Pfad
- ▶ Auschecken

`:pserver:alice@entmm3:/home/cvsmf/test`



Erste Schritte: Auschecken

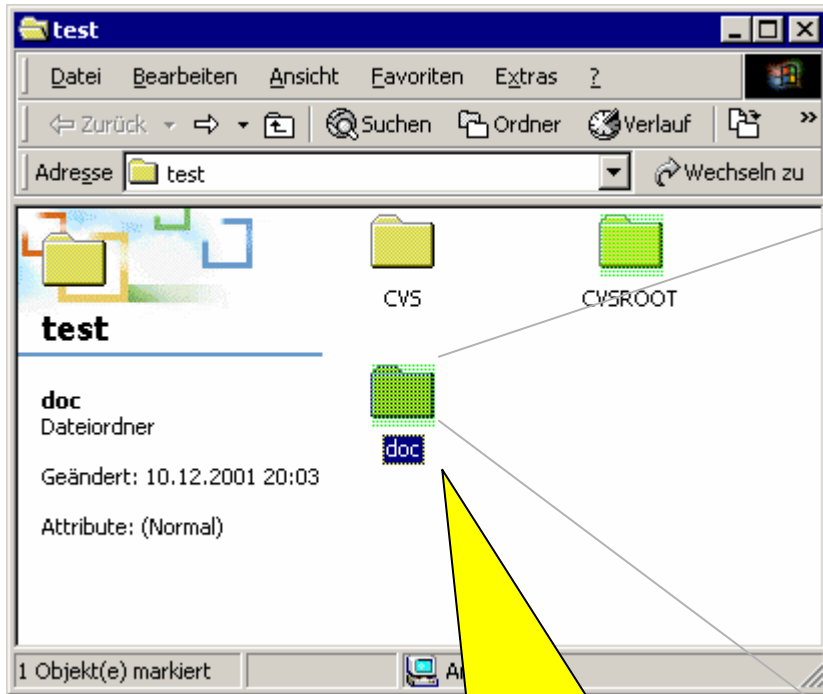
- ▶ Checkout-Operation
- ▶ Ansicht der neuen Dateien

```
TortoiseCVS - Finished
In C:\Dokumente und Einstellungen\cichon\Desktop\test: cvs -q -z6 checkout .
U CVSROOT/checkoutlist
U CVSROOT/commitinfo
U CVSROOT/config
U CVSROOT/cvswrappers
U CVSROOT/editinfo
U CVSROOT/logininfo
U CVSROOT/modules
U CVSROOT/notify
U CVSROOT/rcsinfo
U CVSROOT/taginfo
U CVSROOT/verifymsg
U doc/Makefile
U doc/hello.c
U doc/readme.txt

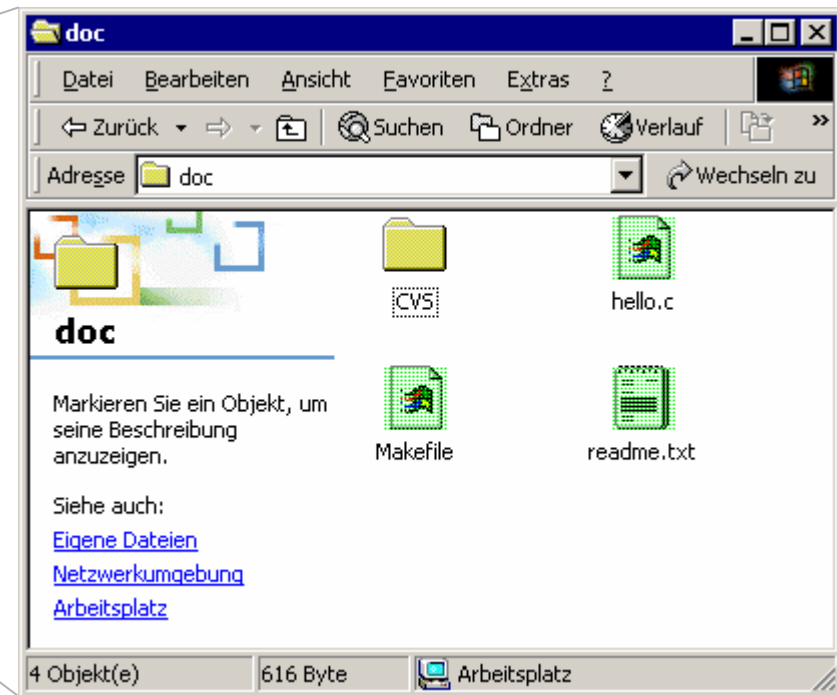
Success, CVS operation completed
```

Close

CVS-Status



grün: unmodifiziert



Erste Schritte: Auschecken unter Unix

▶ Beim ersten Mal: Setzen der CVSROOT-Variable:

- csh

```
> setenv CVSROOT :pserver:alice@entmm3:/home/cvsmf/test
```

- bash

```
> export CVSROOT=:pserver:alice@entmm3:/home/cvsmf/test
```

▶ Beim ersten Mal: Passwort merken

```
> cvs login
```

▶ Auschecken

```
> cvs checkout .
```

Hier ist ein Punkt



CVS im Alltag: Änderungen

- ▶ Ändern der Datei ‚hello.c‘

rot: geändert

Dateiname

Vorher

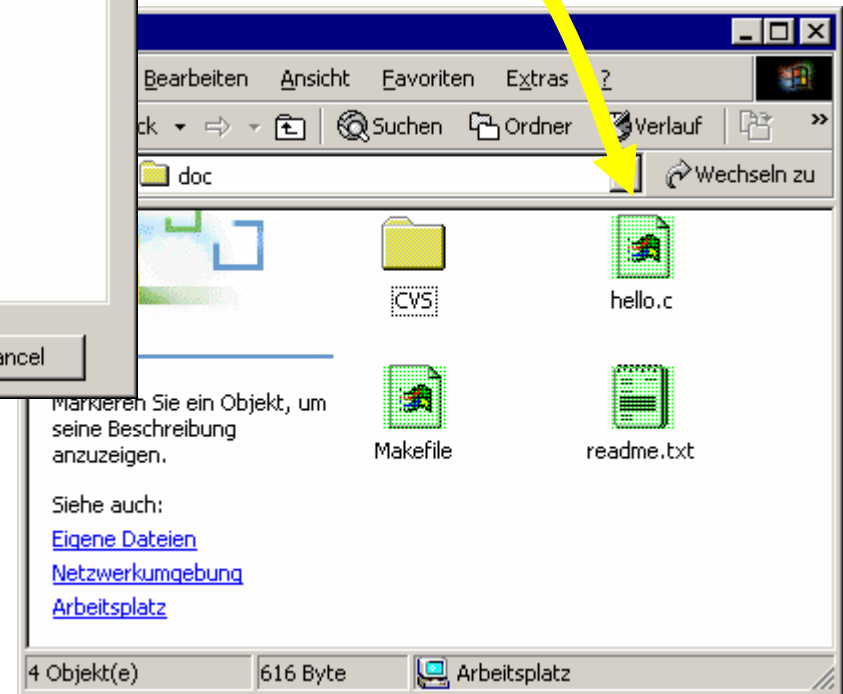
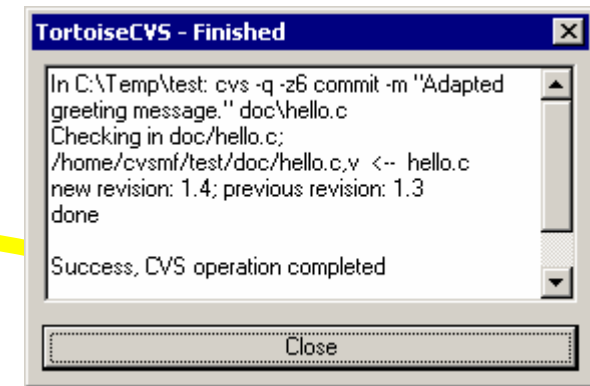
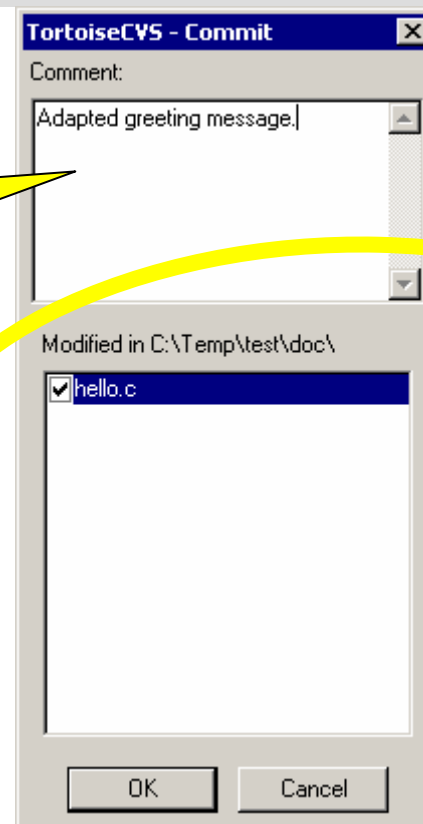
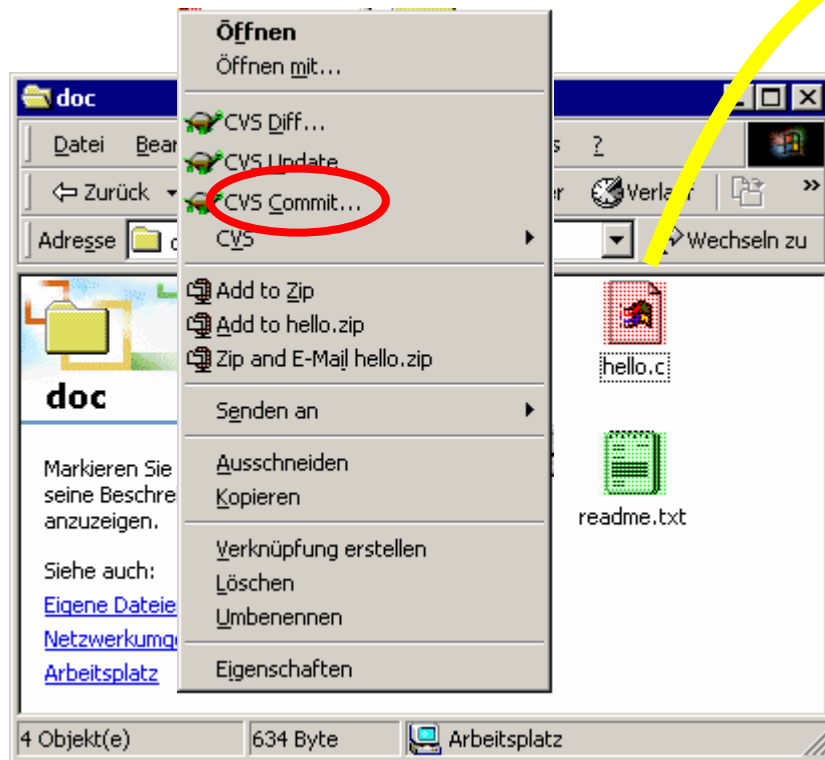
Nachher

The image shows a Windows file explorer window titled 'doc' with a context menu open over the file 'hello.c'. The 'CVS Diff...' option is circled in red. A yellow callout box labeled 'rot: geändert' points to the file icon. Another yellow callout box labeled 'Dateiname' points to the file name. A third yellow callout box labeled 'Vorher' points to the 'CVS Diff...' option. A fourth yellow callout box labeled 'Nachher' points to the diff window. The diff window, titled 'Tortoise CVS - Finished', shows the output of a 'diff' command. The output includes the RCS file path and the diff of 'hello.c'. The diff shows two versions of the file: the first version has a single 'printf' statement, and the second version has two 'printf' statements. The second version's changes are circled in red. The diff window also has a 'Close' button at the bottom.

```
In C:\Documents und Einstellungen\cichon\Desktop: cvs -q -z6 diff
doc\hello.c
Index: doc\hello.c
-----
=
RCS file: /home/cvsmf/tes/doc/hello.c,v
retrieving revision 1.1
diff -c1 hello.c
6c6
< printf ("Hello, world!\n");
---
> printf ("Die CATS-Gruppe ist super-cool!\n");
```


CVS im Alltag: Einchecken von Änderungen

Wichtig: Sinnvolle Beschreibung der Änderung



CVS im Alltag: Änderungen/Einchecken mit Unix

- ▶ Anzeige von Änderungen:

```
> cvs diff
```

- ▶ Einchecken von Änderungen:

```
> cvs commit
```

- ▶ Einstellen des Editors für die Änderungs-Beschreibung

\$EDITOR Umgebungsvariable



CVS im Alltag: Umgang mit Konflikten

TortoiseCVS - Finished

```
In C:\Temp\test: cvs -q -z6 commit -m "Changed greeting from CATS to DVB." doc\hello.c
cvs server: file 'doc/hello.c' had a conflict and has not been modified
cvs [server aborted]: correct above errors first!

Error, CVS operation failed
```

Alte Version

Neue Version

```
hello.c - Editor
Datei Bearbeiten Format ?
#include <stdio.h>
#include <stdlib.h>

int main ()
{
<<<<<< hello.c
    printf ("DVB ist super-cool!\n");
=====
    printf ("Mobilfunk ist super-cool!\n");
>>>>>> 1.3
    return 0;
}
```

CVS im Alltag: Dateien hinzufügen oder löschen

The image shows a Windows Explorer window with a file named 'test.txt' selected. A context menu is open over 'test.txt', with the 'CVS Add' option circled in red. A yellow arrow points from this menu to a 'TortoiseCVS - Finished' dialog box. The dialog box contains the following text: 'In C:\Temp\test: cvs -q -z6 add doc\test.txt', 'cvs server: use 'cvs commit' to add this file permanently', and 'Success, CVS operation completed'. Below the dialog box, a black box contains the command: `> cvs add test.txt`.

Another yellow arrow points from the 'TortoiseCVS - Finished' dialog box to a 'doc' directory in the Explorer window. In the 'doc' directory, a context menu is open over 'test.txt', with the 'Remove' option circled in red. A yellow arrow points from this menu to another 'TortoiseCVS - Finished' dialog box. The dialog box contains the following text: 'In C:\Temp\test: cvs -q -z6 remove doc\test.txt', 'cvs server: use 'cvs commit' to remove this file permanently', and 'Success, CVS operation completed'. Below this dialog box, a black box contains the command: `> cvs remove test.txt`.

A yellow callout box with a black border and a yellow background contains the text 'Unbekannte Datei' (Unknown File) with a yellow arrow pointing to the 'test.txt' file in the Explorer window.

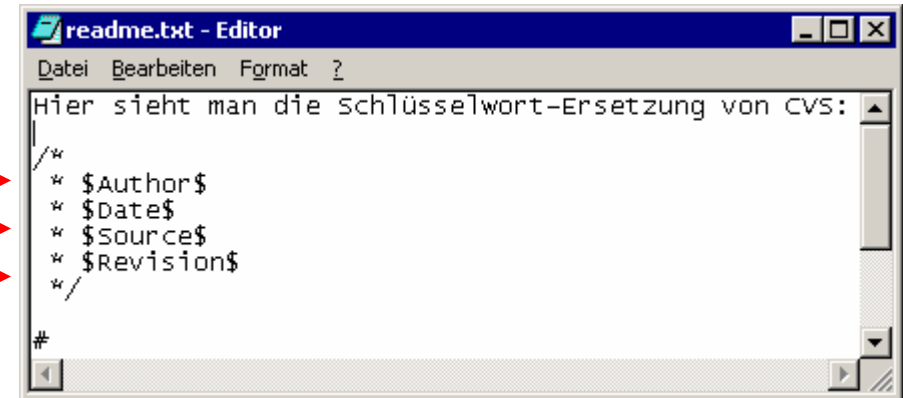
At the bottom right, a context menu is open over the 'CVS' folder, with the 'CVS' option highlighted by a red arrow.



CVS-Schlüsselwörter und Binärdateien

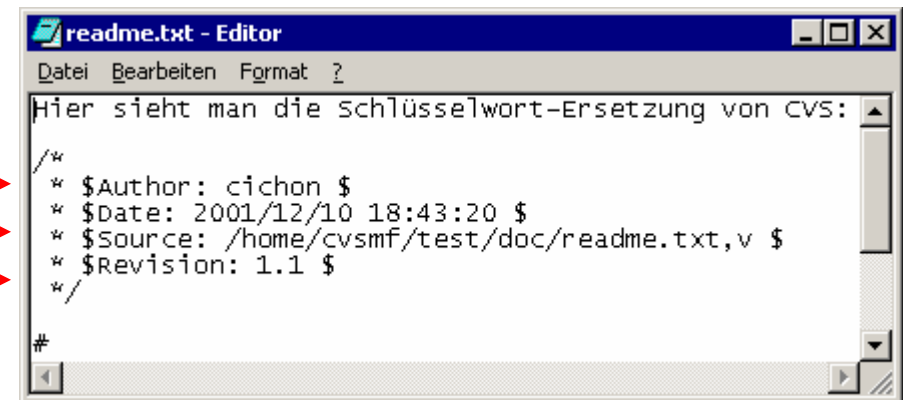
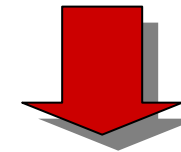
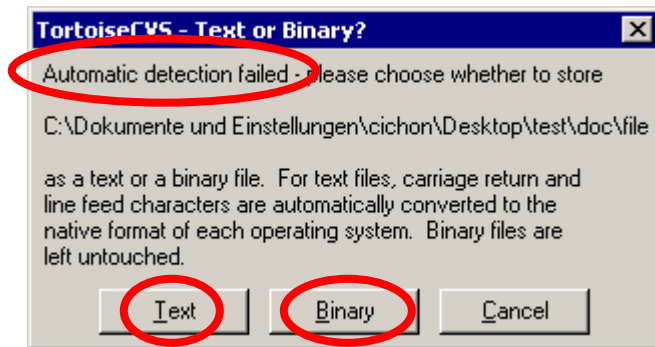
▶ Textersetzungen beim Checkout

- Zeilenende Dos/Unix
- Schlüsselwörter `$Keyword$`



```
readme.txt - Editor
Datei Bearbeiten Format ?
Hier sieht man die Schlüsselwort-Ersetzung von CVS:
/*
 * $Author$
 * $Date$
 * $Source$
 * $Revision$
 */
#
```

Three red arrows point to the keyword substitution lines: `* $Author$`, `* $Date$`, and `* $Source$`.



```
readme.txt - Editor
Datei Bearbeiten Format ?
Hier sieht man die Schlüsselwort-Ersetzung von CVS:
/*
 * $Author: cichon $
 * $Date: 2001/12/10 18:43:20 $
 * $Source: /home/cvsmf/test/doc/readme.txt,v $
 * $Revision: 1.1 $
 */
#
```

Three red arrows point to the substituted keyword values: `* $Author: cichon $`, `* $Date: 2001/12/10 18:43:20 $`, and `* $Source: /home/cvsmf/test/doc/readme.txt,v $`.

```
> cvs add -kb test.txt
```

Wichtig: Angabe unter Unix



Anlegen von Releases

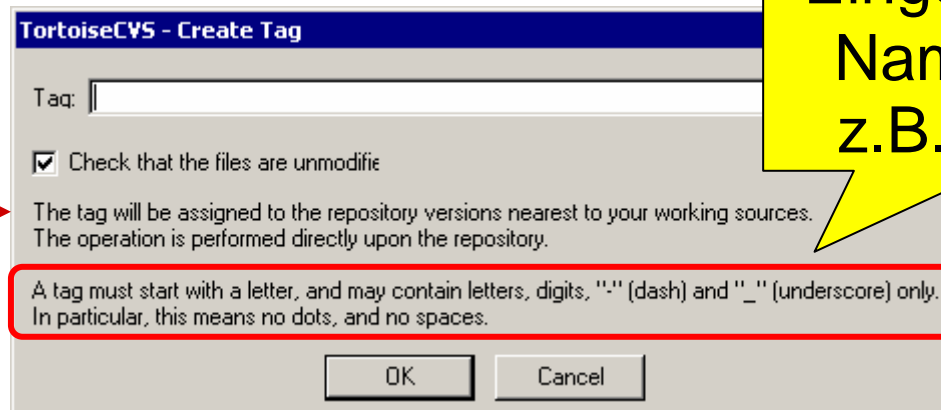
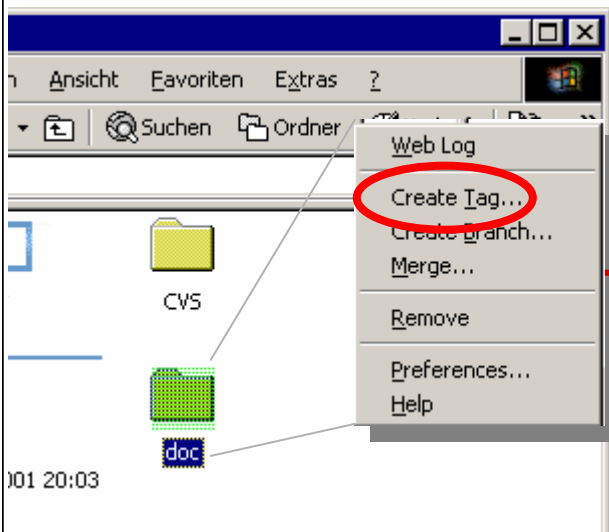
- ▶ Jede Datei erhält bei jeder Änderung eine fortlaufende **Versionsnummer**
- ▶ Ein **Release** besteht aus einer Zuordnung einer bestimmten Version zu jeder Datei.
- ▶ Ein Release erhält einen eindeutigen Namen, genannt **Tag**

Dateien

Zeit

Tag

hello.c	Makefile	readme.txt
1.1	1.1	
1.2		
1.3		1.1
1.4	1.2	
1.5		1.2
1.6		
1.7	1.3	
1.8	1.4	1.3



```
> cvs tag VER-1_0 doc
```

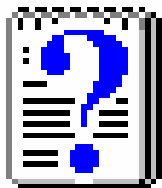


Weitere Informationsquellen

- ▶ **Meine Homepage:** <http://www.ifn.et.tu-dresden.de/~cichon/cvs/>
- ▶ **CVS-Homepage:** <http://www.cvshome.org/>
- ▶ **CVS-Manual (engl.):**
<http://www.cvshome.org/docs/manual/cvs.html>
- ▶ **CVS-Howto:** <http://www.linuxdoc.org/HOWTO/CVS-RCS-HOWTO.html>
- ▶ **Einführung (dt.):** <http://www.koehntopp.de/kris/artikel/cvs/>
- ▶ **Einführung (engl.):**
<http://www.cdt.luth.se/~peppar/presentations/cvs/>



Vielen Dank für das Interesse



▶ Fragen?

- Jetzt, oder
- per E-Mail an cichon@ifn.et.tu-dresden.de

alles-klar.txt

> **cv**s **commit**

