

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



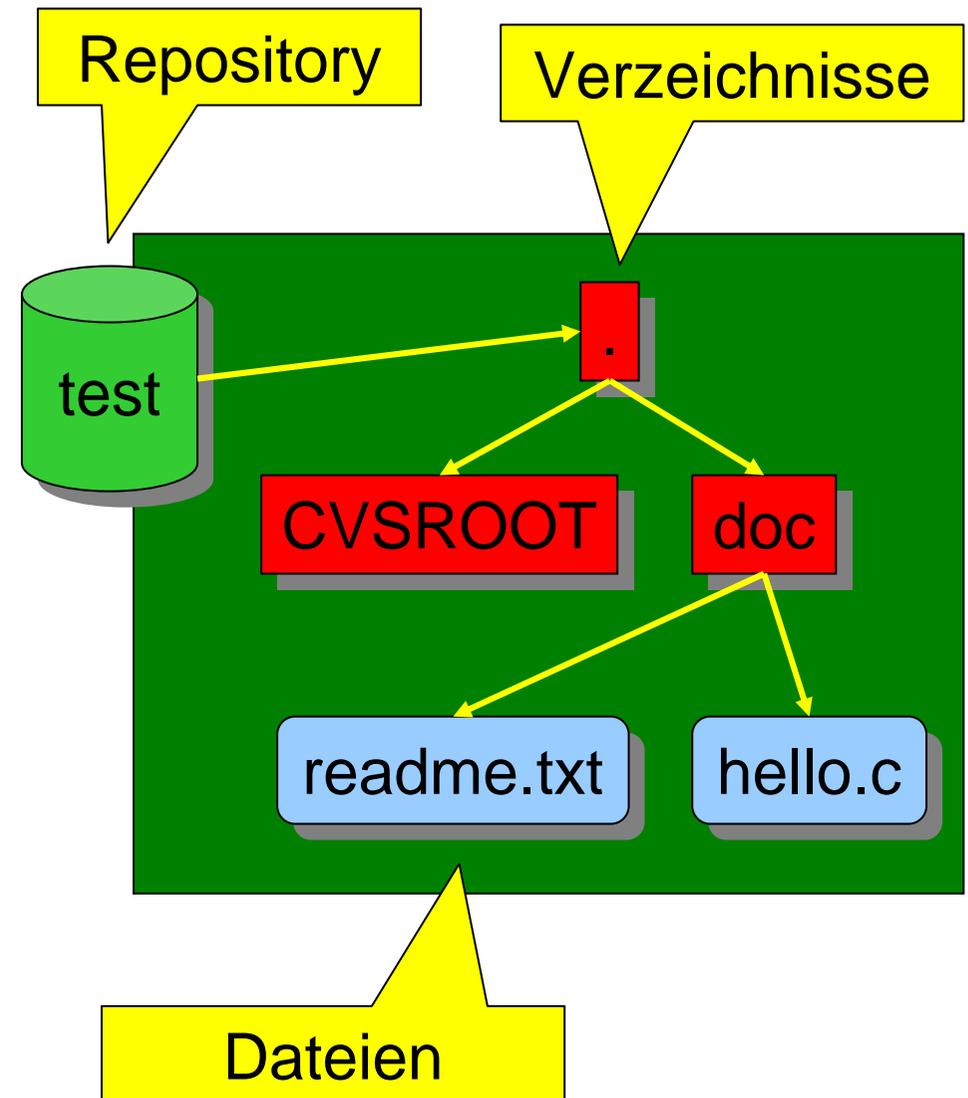
Einführung

- ▶ 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



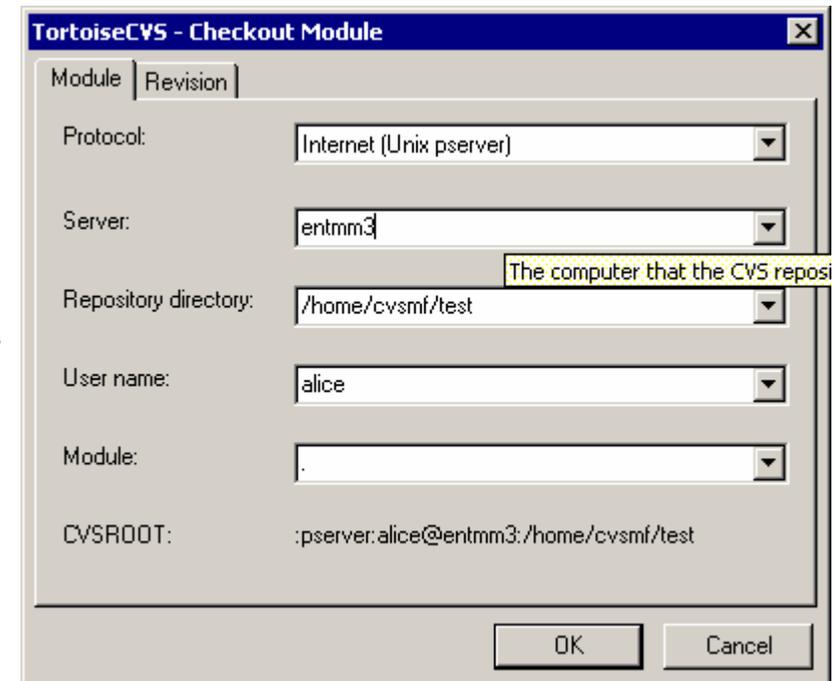
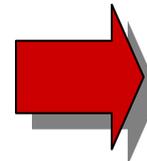
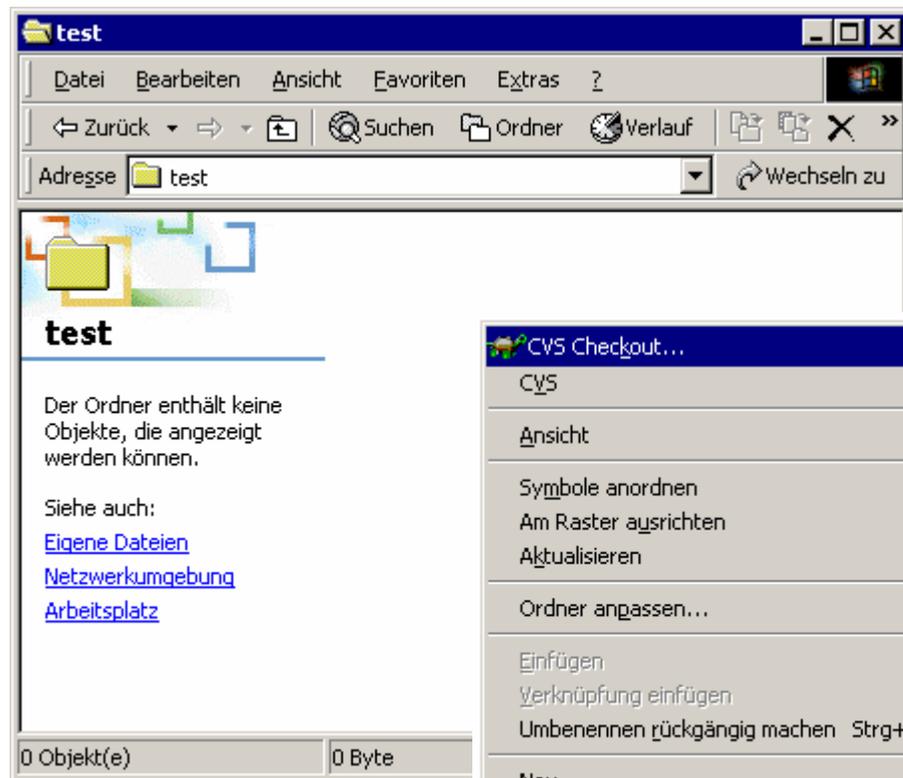
Erste Schritte: Auschecken

Benutzername

Server

- ▶ Voraussetzung: CVS-Repository Pfad
- ▶ Auschecken

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



Erste Schritte: Auschecken

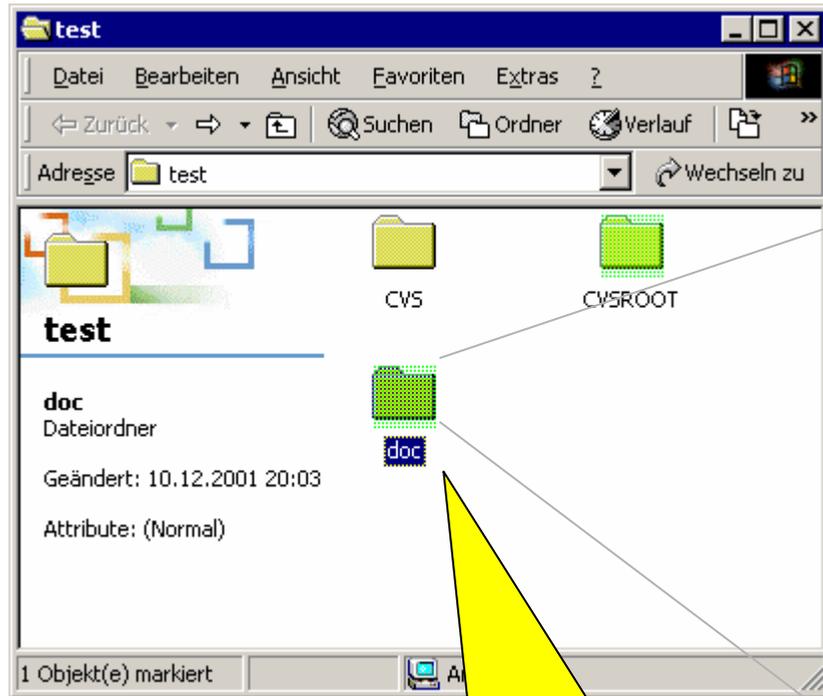
- ▶ Checkout-Operation
- ▶ Ansicht der neuen Dateien

```
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

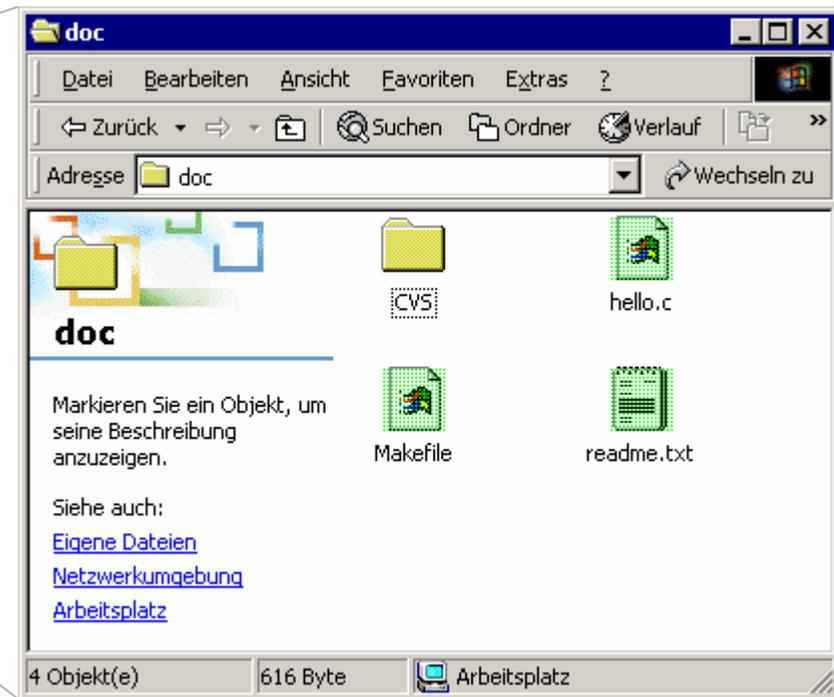
Success, CVS operation completed
```

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‘

The image illustrates the process of checking out a file from CVS and viewing its differences. It consists of two main parts:

- File Explorer:** A window titled 'doc' shows a folder containing 'CVS', 'hello.c', 'Makefile', and 'readme.txt'. The file 'hello.c' is highlighted, and a context menu is open. The 'CVS Diff...' option is circled in red. A yellow callout box labeled 'rot: geändert' (red: changed) points to the file icon. Another yellow callout box labeled 'Dateiname' (filename) points to the file name.
- Diff Window:** A window titled 'Tortoise CVS - Finished' displays the output of a diff command. The text shows the RCS file path and the retrieval of revision 1.1. The diff output is as follows:

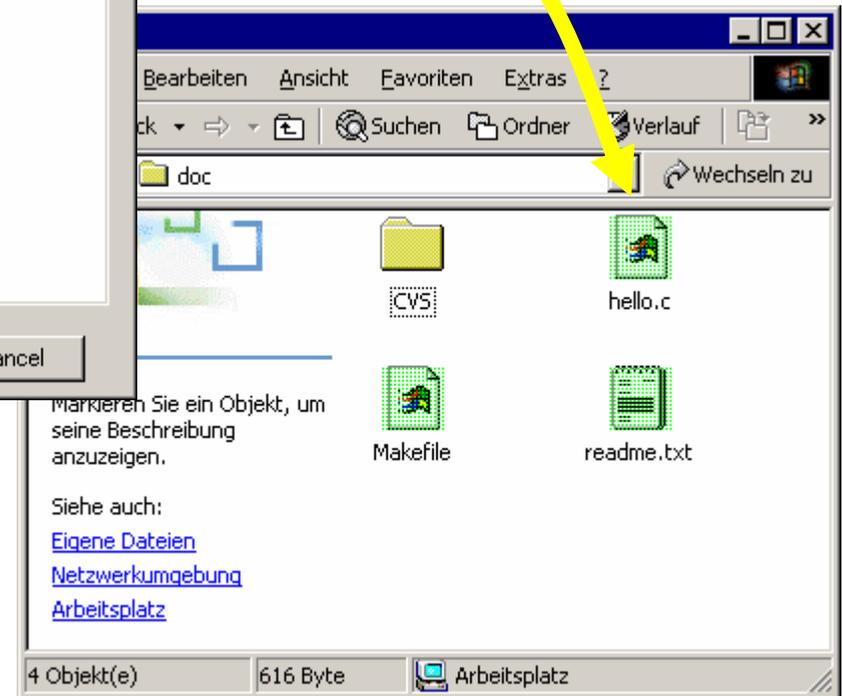
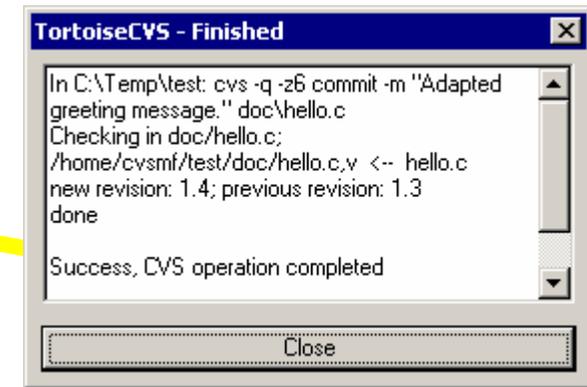
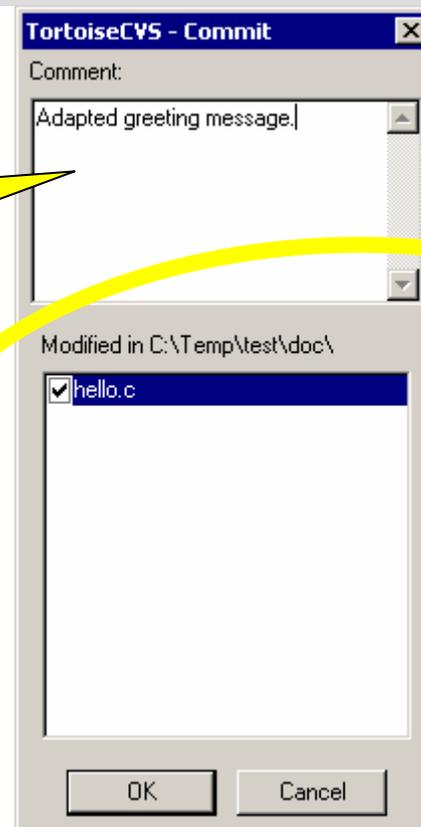
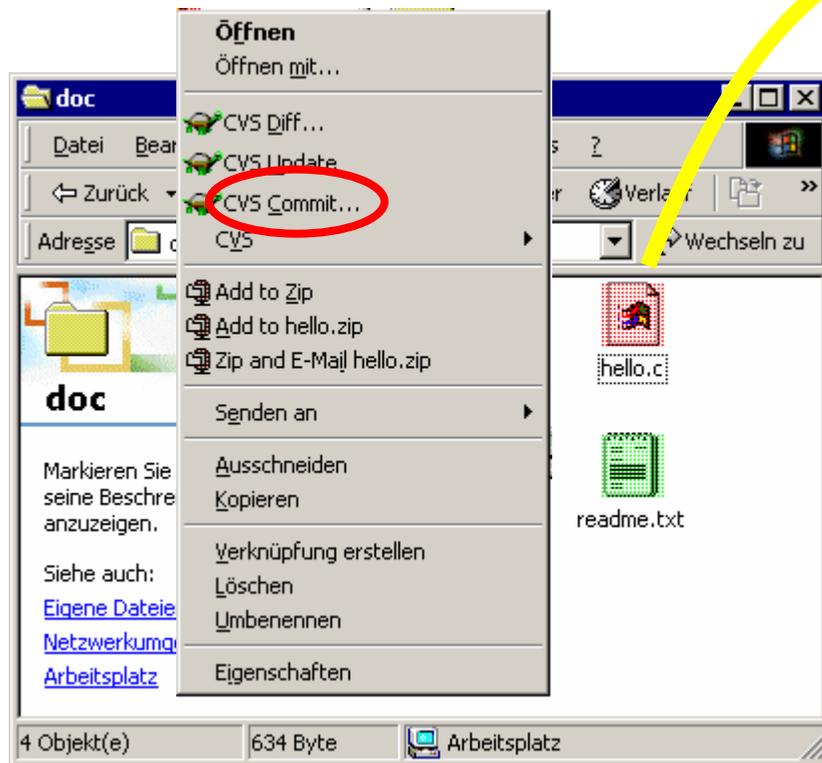
```
In C:\Documents und Einstellungen\cichon\Desktop: cvs -q -z6 diff
doc\hello.c
Index: doc\hello.c
-----
=
RCS file: /home/cvsmf/tes/doc/hello.c,v
retrieving revision 1.1
diff -c .1 hello.c
6c6
< printf ("Hello, world!\n");
...
> printf ("Die CATS-Gruppe ist super-cool!\n");
```

The filename 'hello.c' in the diff output is circled in red. A yellow callout box labeled 'Vorher' (before) points to the first line of the diff, and another yellow callout box labeled 'Nachher' (after) points to the second line. Red arrows point from the 'Ausschneiden' (cut) and 'Kopieren' (copy) options in the context menu to the diff window.



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
```

hello.c - Editor

```
#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;
}
```

Alte Version

Neue Version

CVS im Alltag: Dateien hinzufügen oder löschen

Unbekannte Datei

```
In C:\Temp\test: cvs -q -z6 add doc\test.txt  
cvs server: use 'cvs commit' to add this file permanently  
Success, CVS operation completed
```

```
> cvs add test.txt
```

```
In C:\Temp\test: cvs -q -z6 remove doc\test.txt  
cvs server: use 'cvs commit' to remove this file permanently  
Success, CVS operation completed
```

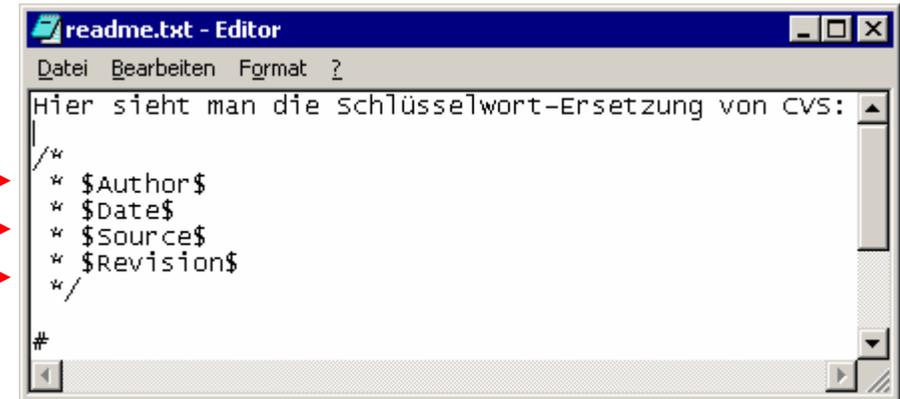
```
> cvs remove test.txt
```



CVS-Schlüsselwörter und Binärdateien

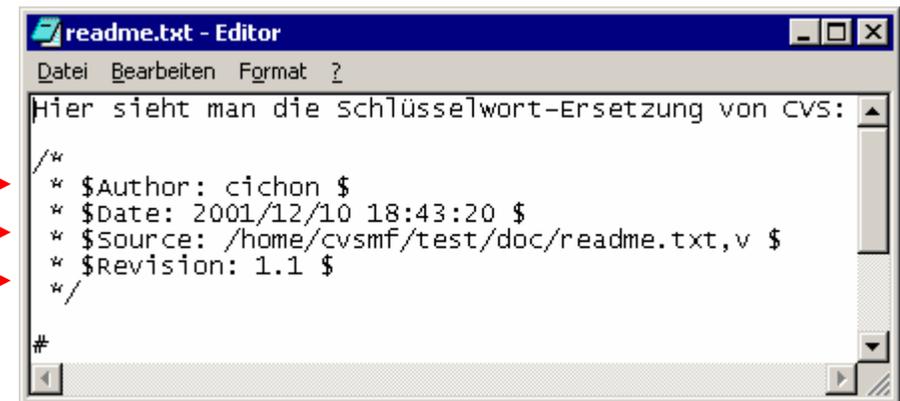
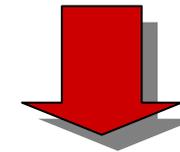
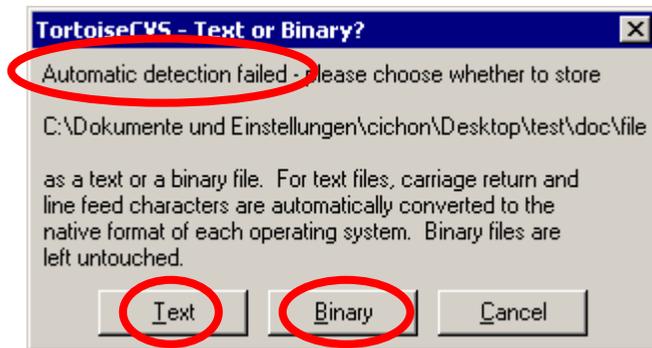
▶ Textersetzungen beim Checkout

- 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 lines: `* $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

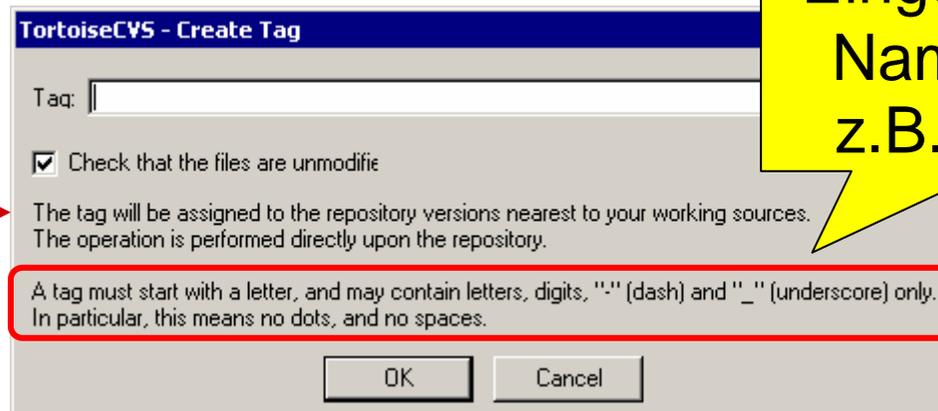
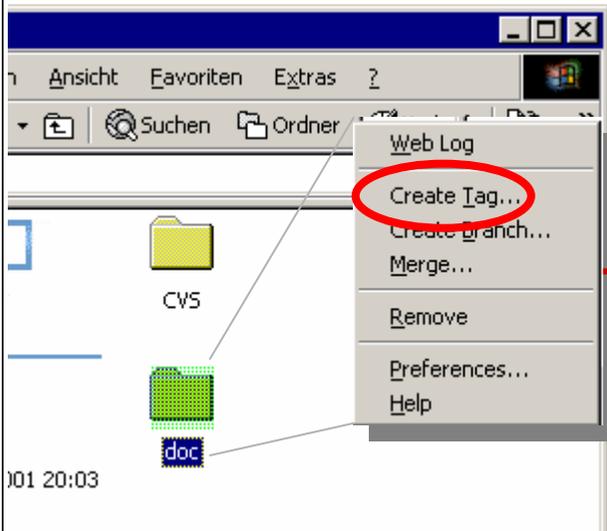
- ▶ Jede Datei erhält bei jeder Änderung eine fortlaufende **Versionsnummer**
- ▶ Ein **Release** besteht aus einer Zuordnung einer bestimmten Version zu jeder Datei.
- ▶ Ein Release erhält einen eindeutigen Namen, genannt **Tag**

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**

