

# ECLIPSE

## In dieser Starthilfe

- Team-Entwicklung mit CVS
  - Anbinden von Eclipse an den CVS Server
  - Einchecken eines vorhandenen Projektes
  - Erzeugen einer Revision
  - Verwalten der Revisionen einer Ressource
  - Ersetzen der aktuellen Revision durch eine andere
  - Projektstand als Version definieren
  - Laden einer Version in den Workspace

## Team- Entwicklung unter Eclipse

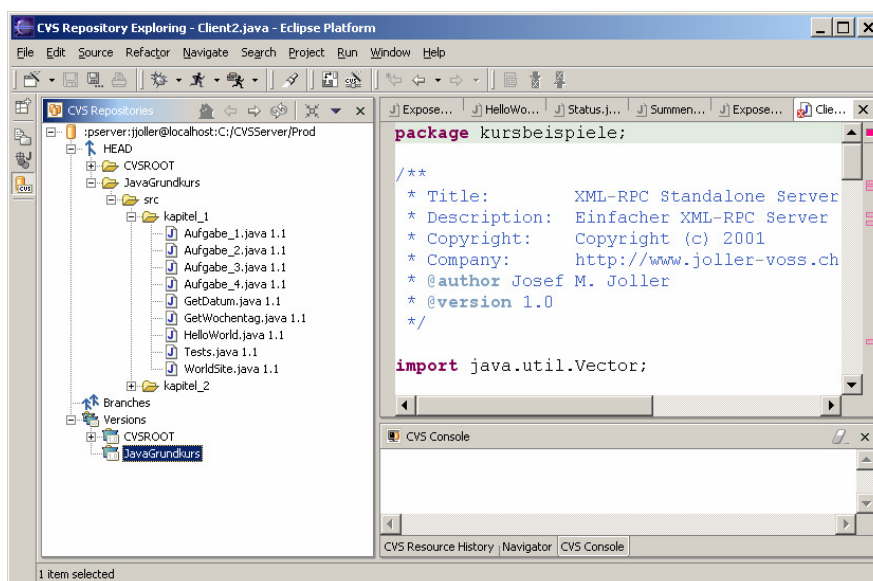
### 1.1. Um was geht's?

Sie wollen sich mit **Eclipse**, einer gratis Java-IDE aus dem *IBM Eclipse Projekt* (Eclipse.org), vertraut machen. Ich wollte das auch und finde Eclipse sehr brauchbar. Einiges finde ich besser als im JBuilder (Refactoring, Plugins, Imports organisieren u.v.m.). JBuilder hat andere Vorteile (Help, UML, grosser Funktionsumfang [Enterprise Edition]). Aber es gibt noch viele andere IDE's, wie beispielsweise IntelliJ ([www.intellij.com](http://www.intellij.com)).

### 1.2. Team-Entwicklung mit CVS

CVS steht für Concurrent Version System. Es ist das bekannteste Open-Source Versionsverwaltungssystem. Sie können beispielsweise Quelldateien verwalten und dabei unterschiedliche Versionen definieren und kontrollieren, ohne dass Sie jeweils den gesamten Code abspeichern müssen. Der Zugriff auf die einzelnen Versionen und Archive sind Passwort-geschützt. Beim Einchecken neuer Dateien kann der Benutzer einen Kurzkomentar angeben und damit die Suche nach bestimmten Versionen erleichtern.

Die Dateien werden in einem Repository (=Verzeichnis mit Unterverzeichnissen) abgespeichert. Jede Veränderung erhält eine Revisionsnummer, wobei mit 1.1 gestartet wird:



# ECLIPSE

(Unsere Dateien befinden sich im Hauptzweig (HEAD) und besitzen die Revisionsnummer 1.1.)

Unter einer Version versteht man den Stand eines Projekts bestehend aus Dateien mit unterschiedlichen Revisionsnummern.

Das CVS hat einen Nachteil: gleichzeitige Änderungen an einer Datei sind möglich. Beim Hineinstellen (*merge*) einer Datei in ein Repository muss der Entwickler entscheiden, wie die verschiedenen Stände kombiniert werden sollen. In der Praxis kann man das Problem so lösen, dass jeder Entwickler seinen eigenen Subpfad verwendet und nur zu bestimmten Zeitpunkten "synchronisiert" wird und eine neue Version im Hauptpfad angelegt wird.

Die Unterstützung für CVS ist direkt in Eclipse integriert, man kann also sofort loslegen. Natürlich muss irgendwo ein CVS-Server zur Verfügung stehen. Zum Testen oder als lokales Repository kann man mit CVSNT (<http://www.cvsnt.org/> für Windows) oder CVS (<http://www.cvs.org/>) schnell einen CVS-Server einrichten. Neben der Möglichkeit im Team an einem Projekt zu arbeiten, sind die wichtigsten Vorteile:

- Änderungshistorie zu jeder Datei
- Änderungskommentare (Begründung jeder Änderung)
- Übersichtliche Änderungsanzeige (was hat sich getan?)
- Verwalten von Tags (Versionsständen oder Releases)
- Verwalten von unterschiedlichen Entwicklungszweigen (Branches)

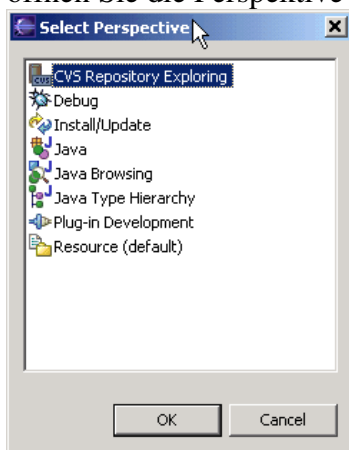
Unabhängig, ob im Team oder nicht gearbeitet wird, sollte man allein aus diesen Gründen immer mit CVS arbeiten.

Weitere Informationen zum CVS (Installation, Tests usw.) finden Sie im *Skript zum CVS* (Starthilfe). Hier beschränken wir uns auf die Nutzung eines CVS Systems aus Eclipse.

## 1.2.1. Anbinden von Eclipse an den CVS Server

Nachdem Sie einen CVS Server aufgesetzt haben (separates Skript, s.o.) möchten Sie sicher Eclipse mit diesem Server verbinden.

1. öffnen Sie die Perspektive "CVS Repository Exploring"



2. Die entsprechende Perspektive wird geöffnet. Selektieren Sie die View "CVS Repositories"  
Wenn Sie jetzt auf die rechte Maus-Taste klicken, erscheint ein Auswahlm Menü

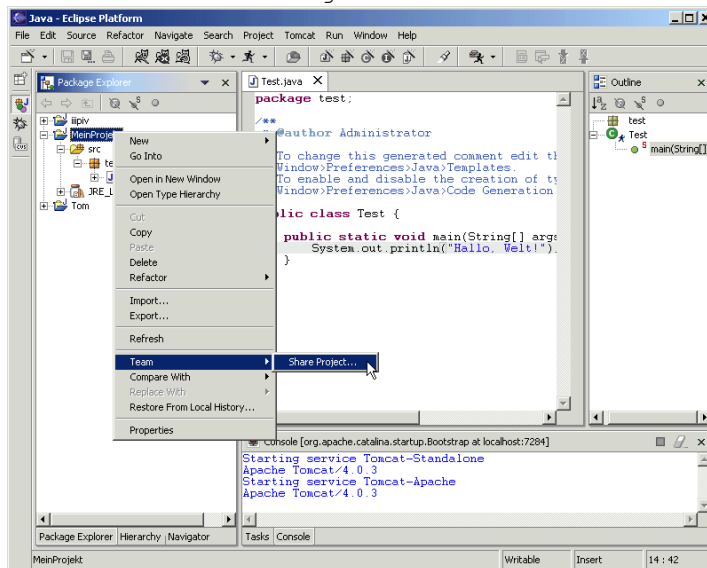


# ECLIPSE

## 1.2.2. Einchecken eines vorhandenen Projektes

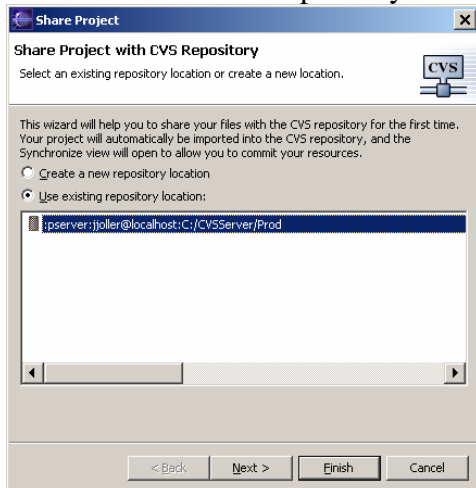
Nun wollen wir aus Eclipse mit CVS Systemen arbeiten. Alle Aktivitäten sind direkt aus Eclipse möglich. Hier das praktische Vorgehen:

1. öffnen Sie eine Perspektive, welche die View "Package Explorer" enthält (oder öffnen Sie in der aktiven Perspektive zusätzlich diese View)
2. Klicken Sie auf das einzucheckende Projekt (selektieren Sie das gewünschte Projekt).
3. Klicken Sie auf die rechte Maustaste. Ein Kontextmenü öffnet sich und u.a. folgende Optionen werden wählbar:  
"New", "Go Into", ... "Team", ... "Properties"
4. Wählen Sie "Team".  
Ein weiteres Kontextmenü öffnet sich und u.a. folgende Optionen werden wählbar:  
falls Sie das Projekt noch nicht ins CVS gestellt haben:  
"Apply Patch" (kann auch fehlen), "Share Project";  
falls Sie das Projekt bereits eingchecked hatten:  
"Synchronize with ...", "Commit", "Tag as Version", ... "Share Project".  
Sie können den Status auch unter den Projekteigenschaften überprüfen:  
nach dem Einchecken sehen Sie unter den Projekt Properties ein Feld "cvs" mit den CVS Server-Informationen.
5. Wählen Sie "Share Project".



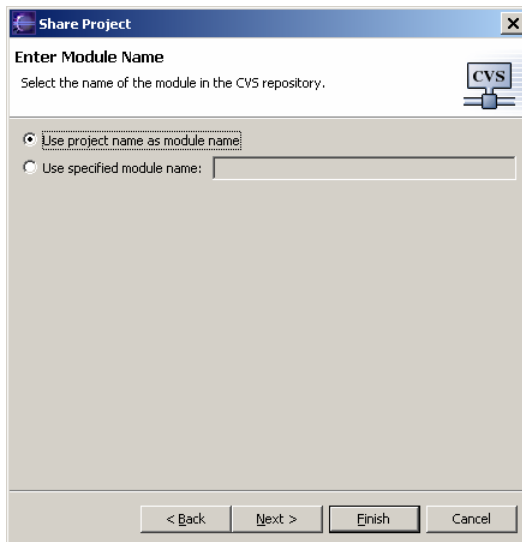
# ECLIPSE

6. Sie können nun das Repository auswählen, in das Sie Ihr Projekt stellen möchten:



Wie Sie sehen, könnten Sie auch gleich ein neues Repository einrichten.

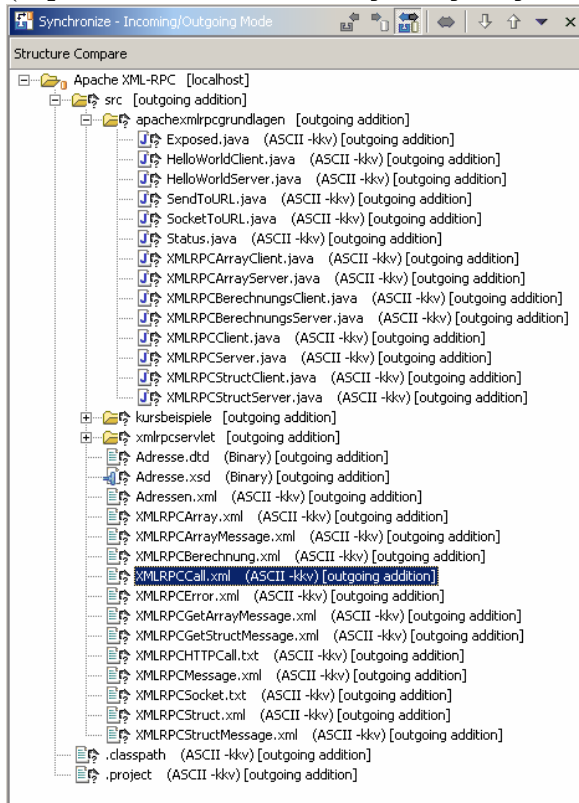
7. Nach Anklicken von "Next" erscheint die folgende Auswahl:



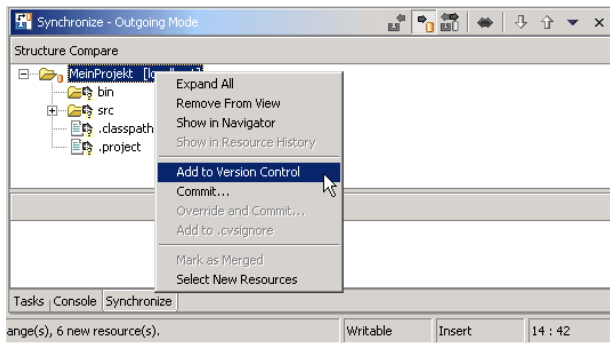
Wenn Sie möchten, könnten Sie Ihrem CVS Modul einen spezifischen Namen geben. Wir lassen die obige Einstellung stehen und gehen zu "Next" und "Finish"

# ECLIPSE

8. Eclipse checkt Ihr Projekt ins CVS ein und Sie können in der CVS View ("Synchronize - Incoming/Outgoing Mode") das Ergebnisprotokoll ansehen:

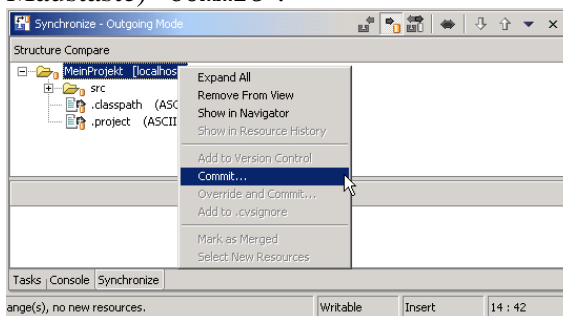


9. In der "Synchronize" View : klicken Sie auf die rechte Maustaste (öffnen Sie das Kontextmenü). Die Optionen "Expand all", "Add To Version Control", ... erscheinen. Wählen Sie "Add To Version Control":



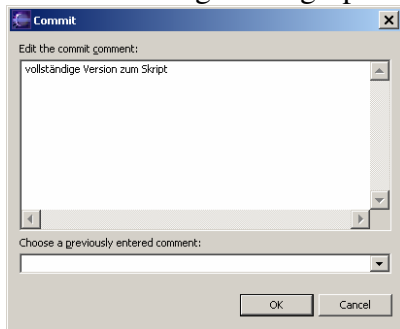
Eclipse fügt Ihr Projekt in die Versionskontrolle ein.

10. Nach einer Weile, je nach Projektgröße: wählen Sie nun aus dem Kontextmenü (linke Maustaste) "Commit".



# ECLIPSE

11. Sie können noch einen erklärenden Text eingeben (dieser sollte in irgend einem Zusammenhang mit abgespeicherten Version sein).



12. Klicken Sie auf "OK". Jetzt werden die Dateien ins CVS übertragen.  
In der View "CVS Repositories" können Sie die eingeecheckten Projekte finden:



wobei die Startversion mit 1.1 bezeichnet wird (die Nummer ist willkürlich – einfach ein Index).

# ECLIPSE

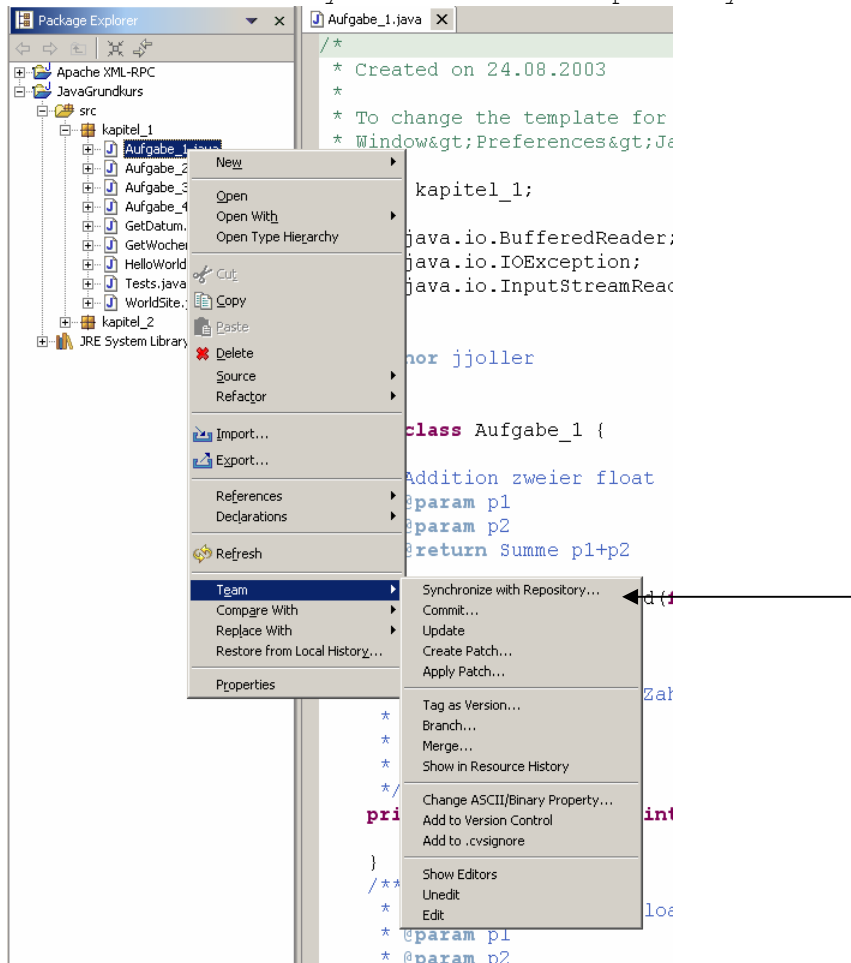
## 1.2.3. Erzeugen einer Revision

Sobald Sie Ihre Dateien / Programme verändern und neu synchronisieren, ergibt sich ein neuer Revisionstand auf dem CVS Server.

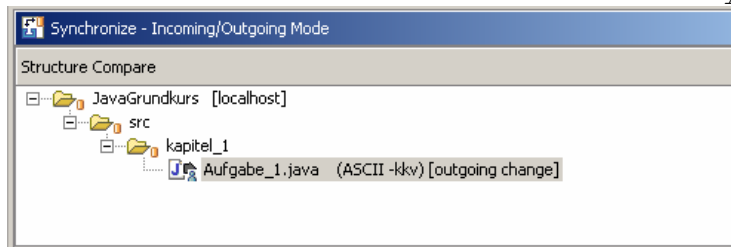
Schritt für Schritt:

1. Ändern Sie eines Ihrer Programme aus einem eingetragenen Projekt (fügen Sie beispielsweise einen Kommentar ein).
2. Speichern Sie die geänderte Datei (Ctrl+S).
3. Jetzt müssen wir die geänderte Datei mit der im CVS synchronisieren.

Kontextmenü: "Team->Synchronize with Repository"



Ihre Datei wird verarbeitet und es erscheint die View "Synchronize"

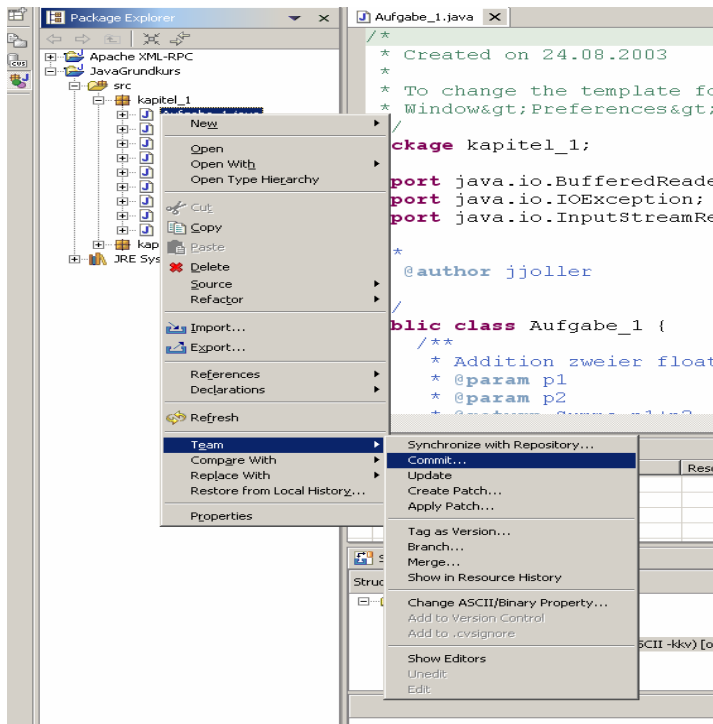


4. Jetzt müssen Sie die Änderung(en) bestätigen.

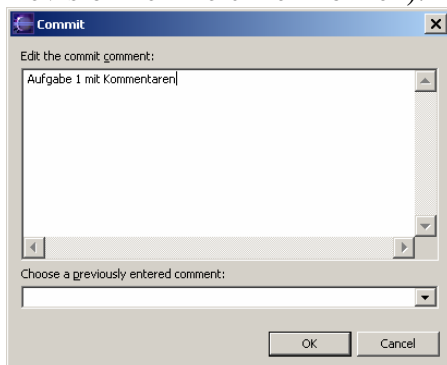
Kontextmenü: "Team->commit"



# ECLIPSE



5. Sie werden aufgefordert einen Kommentar einzugeben (mit dem Sie die aktuelle Revision kennzeichnen können):



Klicken Sie auf "OK".

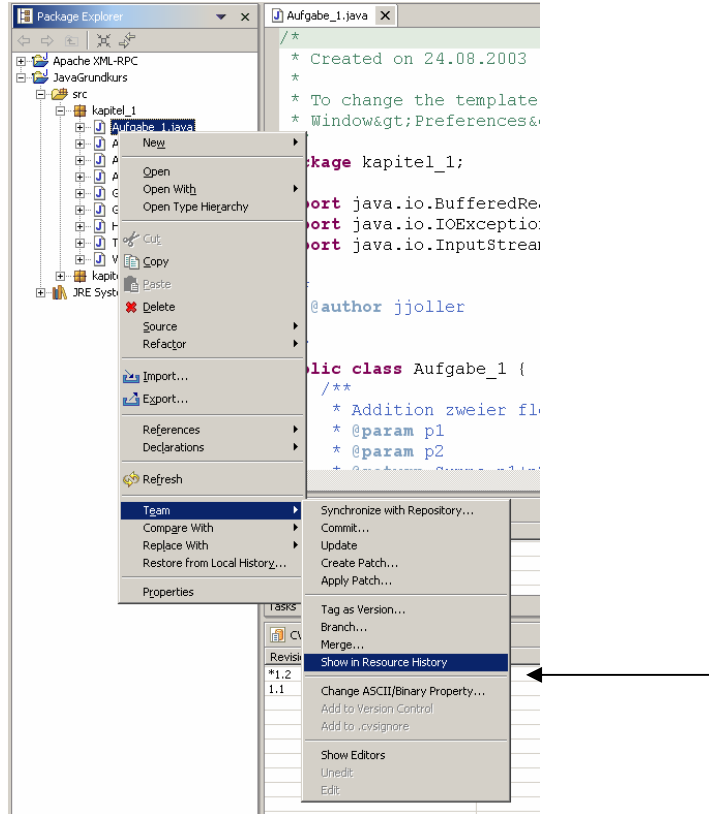
# ECLIPSE

## 1.2.4. Verwalten der Revisionen einer Ressource

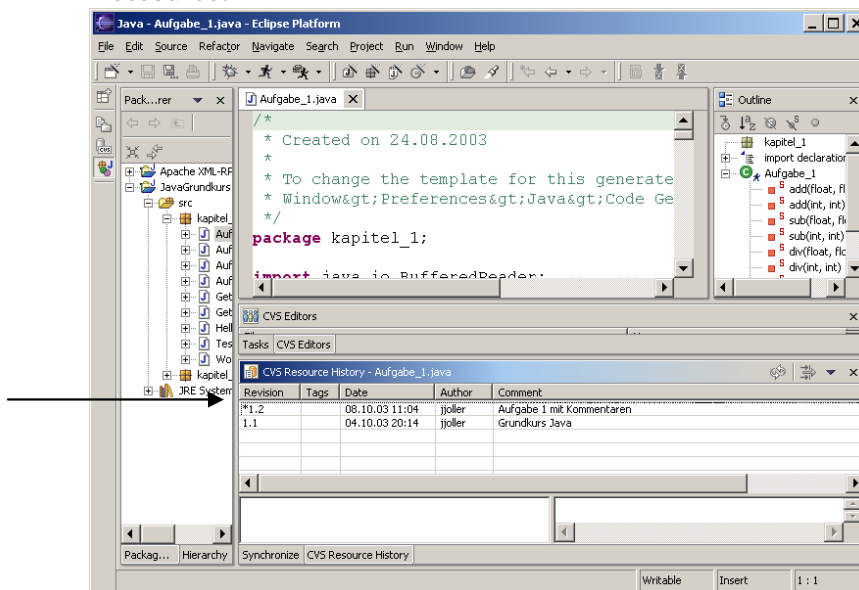
Wenn Sie jetzt sehen wollen, welchen Revisionslevel die gerade mutierte Datei im CVS hat, können Sie wie folgt vorgehen:

1. In der View "Package Explorer" selektieren Sie die neu eingetragene Datei, nicht das Package!

Kontextmenü auf dieser Datei: "Team->Show in Resource History"



2. Die View "CVS Resource History" erscheint und zeigt die oben ausgewählte Ressource:



Die aktuelle Revision ist mit einem "\*" gekennzeichnet.

# ECLIPSE

## 1.2.5. Ersetzen der aktuellen Revision mit einer anderen:

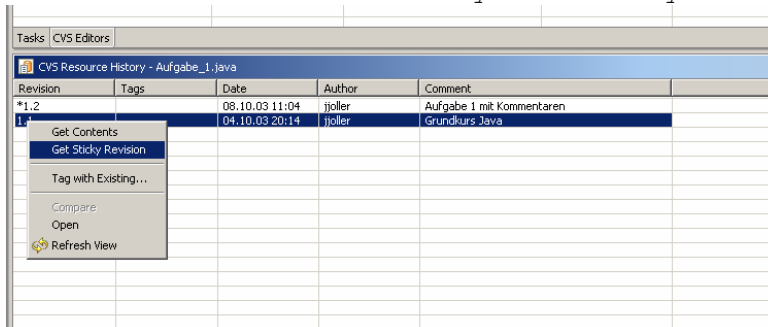
Falls Sie eine Ressource 'versaut' haben und auf eine alte Version zurück greifen möchten, können Sie folgendermassen vorgehen.

1. Selektieren Sie die Ressource (anklicken)
2. Lassen Sie die View "CVS Resource History" anzeigen (jene mit den verschiedenen Revisions).

Kontextmenü: "Team->Show in Resource History".

3. Jetzt können Sie in der View "CVS Resource History" Ihre gewünschte Revision selektieren.

Kontextmenü:"CVS Resource History->Get Sticky Revision"

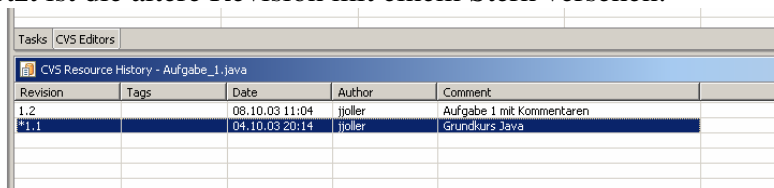


4. Bevor die ältere Revision die neuere *im CVS* als aktuelle ersetzt, erhalten Sie eine Warnung:



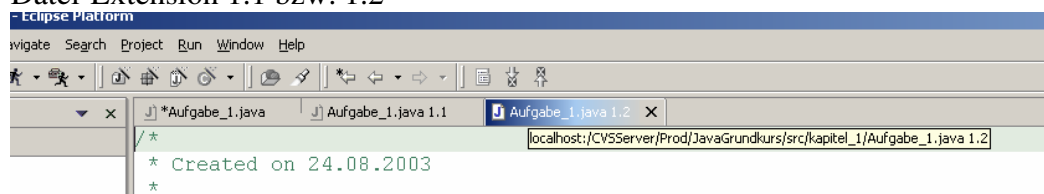
Klicken Sie auf "OK".

5. Jetzt ist die ältere Revision mit einem Stern versehen:



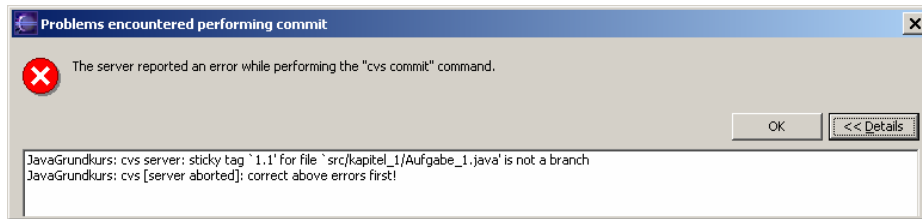
Beachten Sie: bisher haben wir nur CVS-seitig Änderungen gemacht. Im Workspace selber haben Sie immer noch die selbe Revision.

Sie können beide Versionen mit einem Doppelklick im Editor anzeigen lassen, aber mit der Datei-Extension 1.1 bzw. 1.2



Zudem könnten Sie jetzt Probleme bei einer erneuten Synchronisation erhalten:  
EclipseTeamTutorial.doc

# ECLIPSE

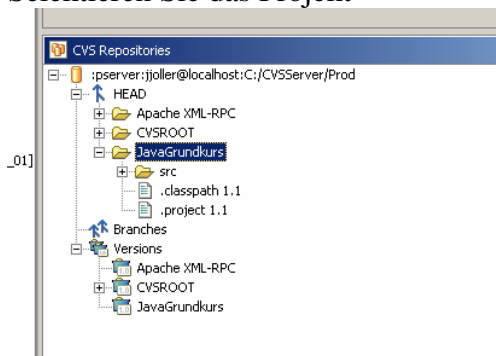


Mit der Revision 1.2 tritt dieses Problem nicht auf.

## 1.2.6. Projektstand als Version definieren

Die Definition einer Version geschieht im CVS. Hier das Vorgehen Schritt für Schritt:

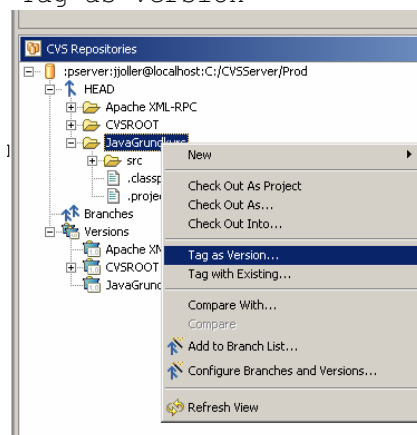
1. Selektieren Sie die View "CVS Repositories"
2. Selektieren Sie das Projekt



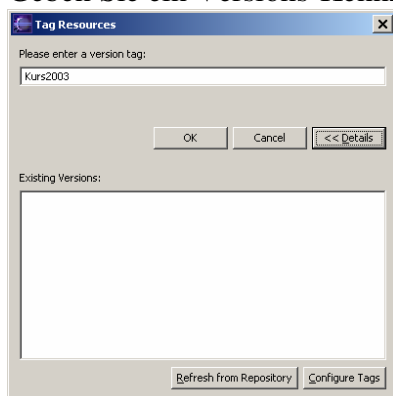
3. Kennzeichnen Sie diese Revision als Version.

Kontextmenü:

"Tag as Version"

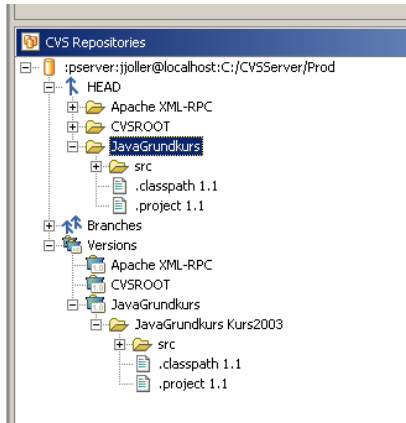


4. Geben Sie ein Versions-Kennzeichen ein, beispielsweise "Kurs2003"



Klicken Sie auf "OK".

5. Jetzt finden wir unter CVS Versionen unsere eben definierte Version:

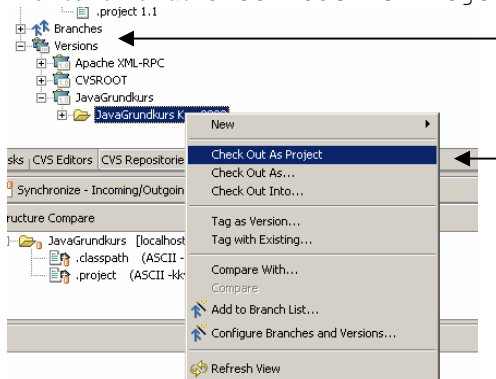


## 1.2.7. Laden einer Version in den Workspace

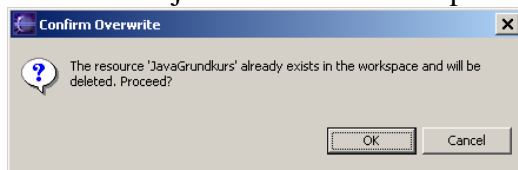
Falls Sie nun Ihre aktuelle, im Workspace befindliche Version durch eine im CVS vorhandene ersetzen möchten, brauchen Sie nur einen Schritt auszuführen.

1. Wählen Sie die View "CVS Repositories"
2. Selektieren Sie die gewünschte Version.
3. Jetzt können Sie die aktuell im Workspace befindliche Version überschreiben.

Kontextmenü: "Check Out As Projekt"



4. Falls das Projekt bereits im Workspace vorhanden ist erscheint eine Warnung:



5. Die im CVS ausgewählte Version überschreibt Ihre Workspace Version.

# ECLIPSE

<b>EINLEITUNG .....</b>	<b>1</b>
1.1. UM WAS GEHT'S? .....	1
1.2. TEAM-ENTWICKLUNG MIT CVS .....	1
1.2.1. <i>Anbinden von Eclipse an den CVS Server</i> .....	2
1.2.2. <i>Einchecken eines vorhandenen Projektes</i> .....	4
1.2.3. <i>Erzeugen einer Revision</i> .....	8
1.2.4. <i>Verwalten der Revisionen einer Ressource</i> .....	10
1.2.5. <i>Ersetzen der aktuellen Revision mit einer anderen:</i> .....	11
1.2.6. <i>Projektstand als Version definieren</i> .....	12
1.2.7. <i>Laden einer Version in den Workspace</i> .....	13