

Upravljanje različic s CVS (Concurrent Versions System)

Peter Peer, peter.peer@fri.uni-lj.si, za XYZ

Spletna stran:

<http://www.cvshome.org/>

Stabilna različica (9.5.2004):

<http://ftp.cvshome.org/release/binary/win32/cvs-1-11-15.zip>

Priročnik:

<http://ftp.cvshome.org/release/stable/cvs-1.11.15/cederqvist-1.11.15.pdf>

Ustvari skladišče:

`cvs -d C:\Progra~1\CVS init`

- v danem direktoriju se ustvari direktorij CVSROOT, ki vsebuje administracijske datoteke CVS
- ostali direktoriji, ki jih ustvarimo z ukazom spodaj, so v istem direktoriju kot CVSROOT in vsebujejo uporabniške module (dejanska skladišča)
- če kdajkoli ponovno zaženemo ta ukaz, se že obstoječe stvari ne prepisejo

Uvozimo projekt, ki ga želimo upravljati (postavimo se v direktorij, ki ga želimo upravljati):

`cvs -d :local:C:/Progra~1/CVS import -m "Uvoz datotek" SecurityAgentADO denali start`

- oznaka *denali* označuje recimo naročnika
- oznaka *start* označuje izdajo
- pozor: uporabljeni so '/'
- če opcije *-m* ni, se zažene urejevalnik, kamor lahko vpišemo željeni komentar ("*Uvoz datotek*"); komentar se vedno zapiše v vsako datoteko skladišča

Sedaj najprej arhivirajmo originalni direktorij in ga nato izbrišimo, da se ne bo slučajno kdaj zgodilo, da bomo popravljali vsebino originalnega direktorija, ki ni vezan na CVS!

Sedaj je skladišče vzpostavljeno in če želimo delati na določenem projektu, moramo najprej dobiti lastno kopijo projekta (postavimo se v direktorij, kamor želimo dati naš delovni direktorij):

`cvs -d :local:C:/Progra~1/CVS checkout SecurityAgentADO`

Ko končamo s spreminjanjem moramo spremembe zapisati v skladišče (smo v našem delovnem direktoriju):

`cvs commit myADO.h`

- ker nismo uporabili opcije *-m*, se zažene urejevalnik, kamor vpišemo komentar revizije, npr.: Dodan optimizacijski prehod: optimizacija berljivosti

Spremembe so sedaj zapisane v skladišče, zato zberemo lastno kopijo (postavimo se v starševski direktorij našega delovnega direktorija):

cvs -d :local:C:/Progra~1/CVS release -d SecurityAgentADO

- ukaz tudi preveri, če smo posredovali v skladišče vse spremembe

Primerjava datotek v delovnem direktoriju s tistimi v skladišču; recimo, da nam v prejšnjem primeru *release* javi, da nismo posredovali vseh sprememb, a se ne spomnimo, kaj smo v podani datoteki spreminjali (smo v delovnem direktoriju):

cvs diff myADO.h

- izpiše nam vse spremembe

cvs status myADO.h

- izpiše le status datoteke

Pregledovanje zgodovine revizij (smo v delovnem direktoriju):

cvs log

- brez parametrov nam ukaz izpiše vse dosegljive informacije o zgodovini revizij (število revizij, datum revizije, avtor revizije, komentar revizije,...)

cvs history -c myADO.h

- ukaz nam izpiše podatke o vsaki spremembi omenjene datoteke v skladišču (podobno kot zgoraj, a bolj kompaktno, vidi pa se tudi delovni direktorij)

cvs annotate myADO.h

- za vsako vrstico datoteke izpiše podatke o zadnji spremembi (revizija, avtor, datum, vsebina)

cvs diff -r 1.1 -r 1.2 myADO.h

- v tem primeru pa nam poda vse spremembe med podanima revizijama

Dostop do kode poljubne revizije (postavimo se v direktorij, kamor želimo dati naš delovni direktorij):

cvs -d :local:C:/Progra~1/CVS checkout -r 1.2 SecurityAgentADO1.2

- če bi želeli spremembe, ki bi jih sedaj naredili na tej reviziji, posredovati v skladišče, nas sistem opozori, da delamo s staro revizijo (uporaba t.i. lepljivih značk); če želimo del te različice prenesti nazaj v najnovejšo različico, lahko naredimo dostopa do obeh revizij in med datotekami prenesemo željene bloke kode

Nadgradnja osnove:

1. Shranjevanje binarnih in ostalih (npr. .obj) datotek
2. Konfiguriranje
3. CVS in GUI
4. V okolju (projektu) večih razvijalcev
5. In še in še...