

Convergence of messaging

CVS

The Converge Group, Olli Pettay
Helsinki 10. syyskuuta 2002
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisältö

1	CVS	1
1.1	Käyttöönotto	1
1.1.1	Alustus	1
1.2	Käyttö	2
1.2.1	Modulit	2
1.2.2	Tiedostot	2
1.2.3	Esimerkki CVS:n käytöstä	3
1.3	Binääritiedostot	3
1.4	CVS:n raportointikirjaimet	4

1 CVS

<suora lainaus>
(CVS-ohje ryhmälle Chromos)

CVS on versionhallintajärjestelmä, eli ohjelmisto, jonka avulla voi arkistoida ja käsitellä ohjelmien tai muiden projektien eri versioita. Versionhallintaa voi tehdä käsinkin, yksinkertaisimmin siten, että jokainen versio talletetaan omaan hakemistoonsa, mutta tällöin muodostuu helposti ongelmaksi tilankäyttö, vaikka jokainen versio pakattaisiinkin. CVS tallettaa vain versioiden väliset muutokset, joten tilankäyttö on huomattavasti tehokkaampaa.

CVS poistaa myös ryhmätyössä usein ongelmallisten samanaikaisten muutosten häviämisen. Usein samaa tiedostoa joutuu muuttamaan useampi ihminen; mikäli he muuttavat tiedostoa samanaikaisesti, aloittaen kumpikin versiosta A, voi toinen kirjoittaa omat muutoksensa, version C, toisen muutosten, version B, päälle, jolloin vain versio C jää jäljelle:

Olli Ohjelmoija:	Hanna Hakkeri:
emacs main.c	emacs main.c
[muuttaa hirveästi]	[muuttaa hirveästi]
[tallettaa tiedoston]	[muuttaa vielä lisää]
	[tallettaa tiedoston]

Koska kumpikin käynnisti editorin samanaikaisesti, jää jäljelle vain sen muutokset, joka tallettaa myöhemmin.

</suora lainaus>

1.1 Käyttöönotto

Jotta CVS:n käyttö olisi yksinkertaisempaa, kannattaa `.profile` -tiedostoon laittaa seuraava määrittely:

```
export CVSROOT=/home/group/converge/cvsroot
```

Näin CVS:lle kerrotaan missä sen keskusvarasto sijaitsee. Mikäli määrittelyä ei tehdä, täytyy jokaisen CVS-komennon yhteydessä kertoa `-d` -optiolla, missä keskusvarasto sijaitsee.

1.1.1 Alustus

Ennen käyttöönottoa keskusvarasto täytyy alustaa komennolla `cvs init`. Alustus tehdään vain kerran. (Eli koska minä olen sen jo tehnyt, ei muiden tarvitse tästä välittää.)

1.2 Käyttö

Tarkemmat ohjeet ks.

```
info cvs  
man cvs
```

1.2.1 Moduilit

CVS säilyttää tiedostoja moduleittain. Käytännössä moduli vastaa alihakemistoja ja niinpä modulin luominenkin tapahtuu seuraavanlaisesti:

```
mkdir moduli  
cd moduli  
cvs import modulinNimi vendorTag releaseTag
```

Modulin nimen valinnan merkitys on siinä, että moduleja ulos otettaessa CVS luo nimeä vastaavan hakemistorakenteen työhakemistoon. VendorTag on jonkinlainen projektin tunnus, jonka paikalla voisi siis käyttää converge -sanaa. ReleaseTag on ensimmäisen version nimi ja sen paikalla käytettäköön sanaa start. Import -komento pyytää käyttäjältä lokimerkintää (sitä pyytää myös moni muu CVS:n komento). Jos merkintää ei anna option -m avulla (esim. cvs import -m "Uusi moduli" testiModuli converge start), käynnistyy jokin editori, jossa käyttäjä voi lokimerkinnän antaa. Käytettävä editori on määritelty ympäristömuuttujissa EDITOR, CVSEEDITOR tai VISUAL.

Kun työskentely modulin tiedostoilla halutaan lopettaa, siirrytään hakemistopuussa yksi ylöspäin (cd ..) ja annetaan komento cvs release -d modulinNimi. Tällöin CVS tutkii onko hakemistossa vielä muutoksia, joille ei commit -käskeyä ole ajettu. Mikäli ongelmia löytyy CVS valittaa niistä käyttäjälle.

1.2.2 Tiedostot

Modulin tiedostojen käsittely onnistuu komennolla cvs checkout modulinNimi, jolloin työhakemistoon syntyy alihakemisto modulinNimi, joka sisältää kaikki modulin tiedostot. Tiedostoja lisätään komennolla cvs add tiedostonNimi, ja jotta lisäys näkyisi myös keskusvarastossa, tulee lisäksi antaa komento cvs commit tiedostonNimi. Commit -käskeyllä päivitetään myös tiedostoihin tehdyt muutokset, mutta ennen sitä kannattaa ajaa komento cvs -n update, joka tutkii onko joku muu tehnyt muutoksia keskusvaraston tiedostoon. Mikäli muutoksia on tehty, tulee käyttää edellistä komentoa ilman -n -optiota, jolloin muuokset päivittyvät myös työhakemiston tiedostoihin.

```
cvs remove tiedostonNimi+cvs commit tiedostonNimi
```

 poistaa tiedoston keskusvarastosta.

```
cvs diff tiedostonNimi
```

 vertaa työhakemiston tiedostoa keskusvaraston tiedostoon ja näyttää tehdyt muutokset.

1.2.3 Esimerkki CVS:n käytöstä

Tässä esimerkissä käyttäjä haluaa muokata projektin dokumentteja, jotka sijaitsevat docs-modulissa.

Ensiksi siirrytään johonkin sopivaan työhakemistoon (joka ei missään nimessä ole cvsroot-hakemiston alla). Tämän jälkeen vaihdetaan ryhmätunnus:

```
newgrp converge
```

Tuodaan moduli CVS:stä työhakemistoon:

```
cvs checkout docs  
cd docs
```

Nyt voidaan tiedostoja muokata. (CVS-hakemistoon ei yleensä kosketa!) Uusia tiedostoja ja hakemistoja voidaan lisätä ja poistaa komennolla

```
cvs add tiedostonNimi  
cvs remove tiedostonNimi
```

Silloin tällöin kannattaa verrata omia tiedostoja CVS-varaston tiedostoihin:

```
cvs -n update tiedostonNimi
```

ja jos joku muu on myös tehnyt muutoksia täytyy ajaa edellinen komento ilman -n -optiota. Tällöin cvs yrittää yhdistää muutokset, mutta mikäli se ei onnistu, cvs tekee #tiedosto.versionimisen kopion käyttäjän tiedostosta ja varsinaisessa tiedostossa on <<<<<<, ===== ja >>>>>> -merkinnöillä kuvattu, missä kohtaa ongelma sijaitsee.

Kun tarvittavat muutokset/korjaukset on tehty annetaan komento

```
cvs commit -m "tähän tulee lokitietoa tarvittaessa" tiedostonNimi
```

"CVS-istunnon" lopetus:

```
cd ..
```

```
cvs release -d docs
```

ja jos haluaa vielä vanhan ryhmätunnuksenkin päälle

```
newgrp
```

1.3 Binääritiedostot

CVS:ään ei yleensä laiteta binääritiedostoja, vaan vain ohjelmien lähdekoodit. Toisinaan binääritiedostoja täytyy kuitenkin käyttää, ja tällöin tiedoston lisääminen tapahtuu komennolla:

```
cvs add -kb tiedostonNimi
```

1.4 CVS:n raportointikirjaimet

update output

'update' and 'checkout' keep you informed of their progress by printing a line for each file, preceded by one character indicating the status of the file:

'U FILE'

The file was brought up to date with respect to the repository. This is done for any file that exists in the repository but not in your source, and for files that you haven't changed but are not the most recent versions available in the repository.

'P FILE'

Like 'U', but the CVS server sends a patch instead of an entire file. These two things accomplish the same thing.

'A FILE'

The file has been added to your private copy of the sources, and will be added to the source repository when you run 'commit' on the file. This is a reminder to you that the file needs to be committed.

'R FILE'

The file has been removed from your private copy of the sources, and will be removed from the source repository when you run 'commit' on the file. This is a reminder to you that the file needs to be committed.

'M FILE'

The file is modified in your working directory.

'M' can indicate one of two states for a file you're working on: either there were no modifications to the same file in the repository, so that your file remains as you last saw it; or there were modifications in the repository as well as in your copy, but they were merged successfully, without conflict, in your working directory.

CVS will print some messages if it merges your work, and a backup copy of your working file (as it looked before you ran 'update') will be made. The exact name of that file is printed while 'update' runs.

'C FILE'

A conflict was detected while trying to merge your changes to FILE with changes from the source repository. FILE (the copy in your working directory) is now the result of attempting to merge the two revisions; an unmodified copy of your file is also in your working directory, with the name ``.#FILE.REVISION'` where REVISION is the revision that your modified file started from. Resolve the conflict as described in `*Note Conflicts example::`. (Note that some systems automatically purge files that begin with ``.#'` if they have not been accessed for a few days. If you intend to keep a copy of your original file, it is a very good idea to rename it.) Under VMS, the file name starts with ``_'` rather than ``.#'`.

``? FILE'`

FILE is in your working directory, but does not correspond to anything in the source repository, and is not in the list of files for CVS to ignore (see the description of the ``-I'` option, and `*note cvsignore::`).