CVS

ja sorsan kunnossapito

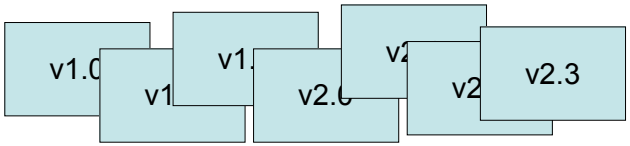
Ongelma

- Missä ja miten säilyttää lähdekoodi niin, että
  - se on kaikkien saatavilla,
  - uusin versio on aina saatavilla,
  - vanhatkin löytyvät.

Jaettu hakemisto

- Idea: yhteinen hakemisto, joka sisältää kaiken lähdekoodin
- Kaikki koodi kaikkien saatavilla
- Entäs kun kaksi henkilöä muuttaa samaa koodia yhtä’ aikaa?
- Mistäs löytyy vanhat versiot?

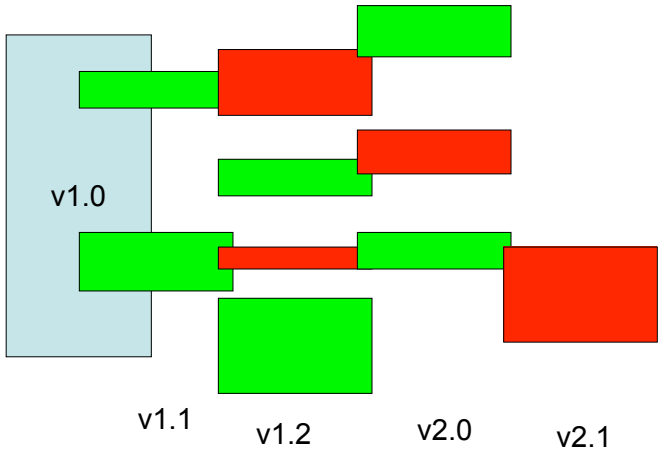
Uudet versiot ilman versiohallintaa



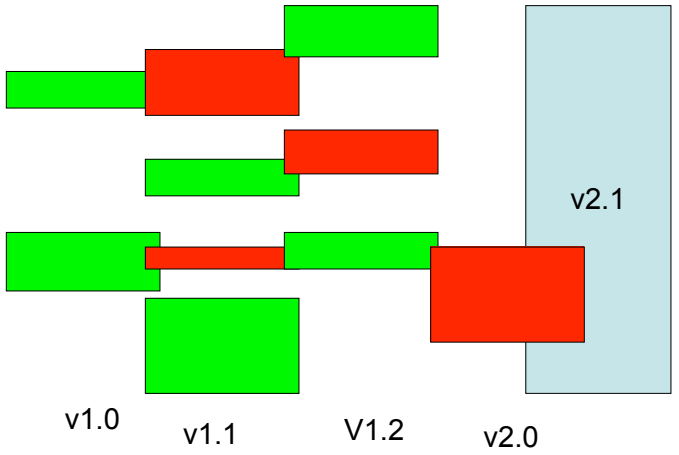
Vanhaa tiedostoa et takaisin saa, ellet sitten tallenna jokaista tiedostoa talteen. Tällainen toiminta olisi kyllä hankalaa ja tarpeetonta.

Versioiden tulisi säilyä

Säilötään vain muutokset



…mutta käänteisesti



Entäs samanaikaiset muutokset?

- Joko kielletään:
  - muokattava tiedosto lukitaan
  - vain yksi kehittäjä muokkaa yhtä tiedostoa kerrallaan
- Tai sallitaan:
  - konfliktitilanteita sattuu harvoin
  - merkitään konfliktit, eikä jatketa, ennen kuin ne on korjattu

Konfliktiesimerkki

```
Kissa istuu
<<<<<<<
puussa
=======
kuussa
>>>>>>>
ja syö juustoa.
```

Oma, paikallinen versio

Muualta tullut versio

Kuis käytetään?

1. Tallennetaan ensimmäinen versio
2. Haetaan muutokset
3. Muokataan
4. Tallennetaan muutokset
5. Toistetaan kohdasta 2., kunnes sovellus valmis

Alustus

- Alustus (vain kerran):
  - Aseta ympäristömuuttuja CVSROOT
    - `CVSROOT=/polku/cvs/hakemistoon`
    - `CVSROOT=`
      `:ext:mie@sorsa.it.jyu.fi:/var/opt/cvs/projekti`
  - aja komento: `cvs init`

Ensimmäinen tallennus

- Hakemistossa, jossa sorsa sijaitsee:
- Aja komento:
  `cvs import -m 'Eka tallennus' pn tn st`
  - pn - projektin nimi
  - tn - oma käyttäjätunnus
  - st - alkutunniste, yleensä *start*
- Ja heti alussa viestit järkeviä EIKÄ IKINÄ tyhjiä!!!
- Windows ' → "

Eka haku

- Poistu projektihakemistosta
- Muuta projektihakemiston nimi, esim. proj → proj.vanha
- Aja komento:
  `cvs checkout` projektinimi
- Voit myös lyhentää checkout → co

Eka tallennus

- Aja komento:
  `cvs commit -m 'Muutoksen kuvaus' tt`
  missä:
  - tt - muuttuneet tiedostot tai piste
- Muista järkevä kommentti!
- Ei tyhjää kommenttia!
- commit lyhenee ci

Muutosten haku

- Aja projektihakemistossa komento:
  `cvs update`
- Jos tiedostoja poistettu, lisää liput `-dPA`, tai aja checkout
- HUOMAA! Update on ajettava *aina* ennen kuin muokkaat tiedostoja tai viet niitä varastoon.

Uudet tiedostot

- Ensin:
  `cvs add -m 'viesti' tiedostot`
- jos binääritiedosto, lisää -kb ensimmäiseksi lipuksi
- Sitten:
  `cvs ci -m 'viesti' tiedostot`

Jotta salaus toimisi

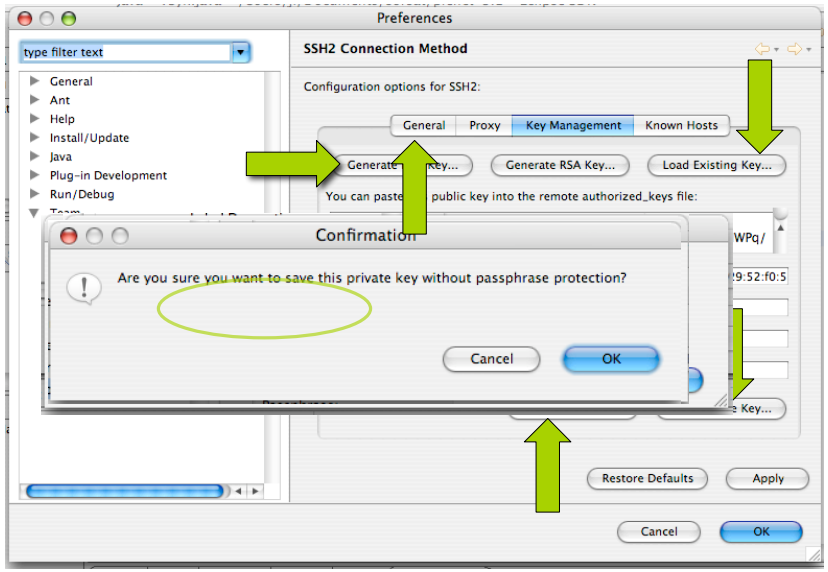
- Avainpari on luotava: `ssh-keygen`
- Julkinen avain on kopioitava koodivarastokoneeseen:
  `scp oma.avain mie@sorsa:.ssh/authorized_keys`
- toki onnistuu ilmankin, mutta on ikävä kirjoitella jatkuvasti salasanaa
- SSH:n ssh-agent toinen vaihtoehto
- ...ja VPN

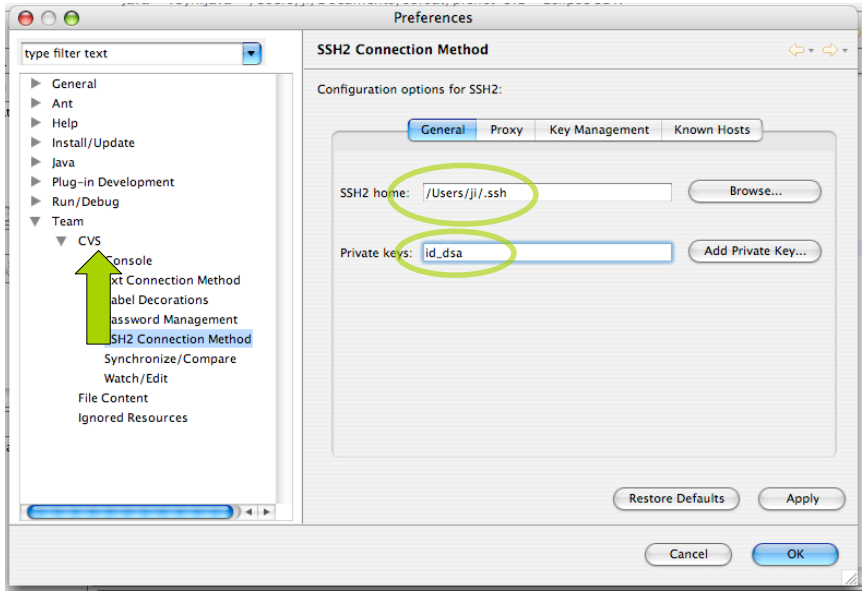
Tietoturva

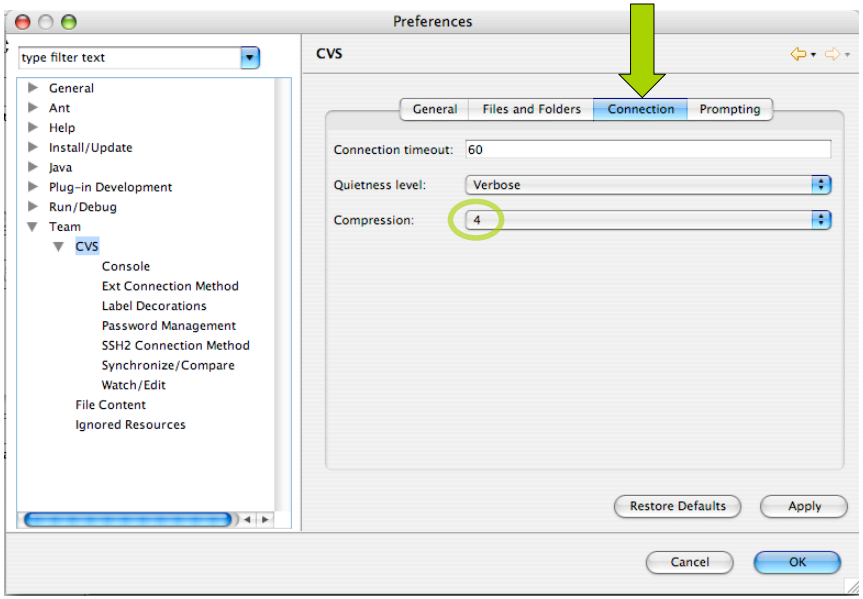
- Ilkeä kräkkeri vaanii joka yhteyden välissä, mistä turva?
- SSH:
  - luodaan salattu putki työkoneen ja tietovaraston välille
  - salaus hoidetaan asymmetrisellä salauksella, eli julkinen avain salaa, yksityinen purkaa

Niin kuinkas…

- …avainparin loisin
- Linux:
  http://www.cc.jyu.fi/~mmhilleb/avaimet.html
- Windows:
  http://www.mit.jyu.fi/atk-tuki/ssh.html
- …tai Eclipse:







Mitä muuta hyötyä CVS:stä?

- Diff:
  - muutokset tallennetaan erotuksina
  - kahta versiota helppo vertailla
- Log:
  - muistatte kirjoittaa ne kommentit aina
  - jälkeenpäin voi katsella, mitä on tullut tehtyä
  - lokista voi päätellä myös kaikkea muuta mukavaa