WhyCvsDir

```
From a mail of Sam Ruby  On 09/05/2003 19.50, to gump-at-jakarta-dot-apache-dot-org
```

Let's look at the initial use case for Gump... a batch and overnight build all.

First day a clean checkout is done. Imagine a build is done in that directory. Now what do we do on day two?

1) We could rm -rf followed by a clean cvs checkout.

2) We could have done the initial checkout to another location, and then synchronize the two. There are commands like rsynch which do this very efficiently, even when the source and target are on the same machine.

Option 2 has a number of secondary advantages. First is that because a cvs update is done on the second day, you can actually get a list of what files have changed. This is a very useful report to have when you find a failure or want to know if your change that you committed at the last minute made it in.

Another advantage is the one that Stephan mentioned. By having a clean checkout local, you can do a lot more experimental changes on your build copy always knowing that you can resynch back toor easily diff against the point where you last checked out (which may be different than the current state of the CVS repository).

Finally, a cvs update is often more bandwidth friendly than a complete checkout.

Now as to what I would consider common, day to day, interactions:

1) resynch with cvsdir (i.e., wipe out my changes)

2) update basedir against the cvs repository (i.e. merge)

That's it! Updating cvsdir is generally best done in batch, or can be relegated to a menu option.

- Sam Ruby