UsingLibrariesInCVS

Including 3rd party code in CVS

Let's say we want to import the excellent database abstraction layer ADOdb version 3.60 into our CVS repository:

get it

```
$ wget http://phplens.com/lens/dl/adodb360.tgz $ tar xvzf adodb360.tgz $ rm adodb360.tgz $ cd adodb
$ cvs import -m 'Imported ADOdb 3.60' _adodb PHPLENS_COM R3_60 $ cd .. $ rm -fr adodb
```

Now, we're going to check it out from CVS and fix a bug we found:

```
$ cvs checkout _adodb $ cd _adodb <i>...hack, chop, whittle...</i> $ cvs commit -m "Fixed bug
#12345: Replace doesn't use native REPLACE command, if available" $ cd .. $ rm -fr _adodb
```

Now, we want to upgrade to version 3.70:

```
$ wget http://phplens.com/lens/dl/adodb370.tgz $ tar xvzf adodb370.tgz $ rm adodb370.tgz $ cd adodb
$ cvs import -m 'Imported ADOdb 3.70' _adodb PHPLENS_COM R3_70
```

This command completed successfully, but reported the following:

```
1 conflicts created by this import. Use the following command to help the merge: cvs checkout -j -
jR3_70 _adodb
```

So, let's delete the imported directory:

```
$ cd .. $ rm -fr adodb
```

And checkout as instructed above. You will find out the prev_rel_tag by looking at a file in sourceforge http://tikiwiki.cvs.sourceforge.net/tikiwiki/_adodb

```
$ cvs checkout -jR3_60 -jR3_70 _adodb $ cd _adodb $ grep -r '<<<<<<' *
```

Manually resolve any conflicts that were reported.

Now, let's commit our 3.60 changes into 3.70:

```
$ cvs commit -m 'Merged our 3.60 changes into 3.70'
```

And finally remove our directory:

```
$ rm -fr _adodb
```

---

merge to branch

All the above being done on HEAD, it can be convenient to cascade the upgrade on branches. Usual tiki merges occur usually from branch to HEAD, now it's the contrary for such third party libs.

```
$ cvs co -r BRANCH-1-9 _adodb $ cd _adodb $ cvs up -jHEAD
```

Then fix conflicts and commit

```
$ grep -r '<<<<<<<' * # .. hack and fix $ cvs ci -m'applied tiki former changes to adodb 3.70'
```