

Cours 6 : Gestion de versions avec `cvs`

Le programme `cvs` (*Concurrent Versions System*) est un outil d'aide à la gestion des versions concurrentes d'un projet. Il est basé sur une architecture client/serveur. Cette fiche rappelle les principales commandes utilisées pour administrer vos projets tant du côté serveur que du côté client.

1 Côté serveur

Création du *Repository*

```
mkdir CHEZMOI/cvs      Attention CHEZMOI doit être un chemin absolu!  
cvs -d CHEZMOI/cvs init  Initialisation du nouveau repository cvs
```

Importation d'un projet

```
cd ~nom/projet      Les fichiers sources de votre projet  
                    de compilation sont ici
```

```
cvs -d CHEZMOI/cvs import projet nom new  Pour créer une nouvelle entrée  
                                           dans le repository cvs et y importer  
                                           les fichiers du répertoire ~nom/projet  
                                           Après cette commande, vous pouvez  
                                           effacer le répertoire contenant les  
                                           sources (~nom/projet).
```

2 Côté client

```
cvs -d CHEZMOI/cvs co projet  Pour créer un répertoire de travail projet contenant  
                               une copie de l'entrée projet du repository cvs
```

Vous devez vous trouver dans votre répertoire de travail pour les commandes suivantes.

Synchronisation avec le *repository*

```
cvs up      Mise à jour du répertoire de travail  
cvs up -d  Mise à jour en descendant récursivement dans les répertoires
```

Ajouter un fichier au *repository*

```
cvs add fichier      Pour ajouter un fichier texte ou un répertoire  
cvs add -kb fichier  Pour ajouter un fichier binaire
```

Supprimer un fichier du *repository*

`rm fichier` Supprimer le fichier de votre répertoire de travail
`cvcs remove fichier` Puis le supprimer du *repository*



Après ajout ou suppression de fichiers dans votre espace de travail à l'aide des commandes `cvcs add/remove`, il faut valider ces changements avec `cvcs commit` pour qu'ils prennent effet dans le dépôt central.

Validation des changements

`cvcs commit -m "commentaire"` Propage les modifications locales au dépôt central

Examiner le dépôt local

`cvcs status` Affiche le statut des fichiers du répertoire courant
`cvcs diff <fichier>` Affiche les différences entre un fichier local et le dépôt central
`cvcs log <fichier>` Affiche toute l'historique d'un fichier

A l'aide !

`cvcs <commande> --help` Affiche de l'aide sur la commande considérée

3 Codes de modification de fichiers

Lorsque vous effectuez une commande de mise à jour `cvcs up`, `cvcs` affiche une liste de fichiers précédés d'un code spécifiant quelle type de modification a été effectuée ou constatée. Les différents codes les plus courants et leur signification sont rappelés ici :

Code	Signification
U	Les modifications du dépôt ont été appliquées à ce fichier, qui n'avait aucune modification locale.
G	Les modifications locales de ce fichier et celles du dépôt ont été fusionnées avec succès.
C	Les modifications locales de ce fichier et celles du dépôt n'ont pas pu être fusionnées : elles sont en conflit.
M	Ce fichier a des modifications locales qui n'ont pas encore été propagées au dépôt.
A	Le fichier a été ajouté au dépôt.
R	Le fichier a été enlevé du dépôt.