



## Jeu d'outils de livraison SVN de APL

Ce jeu d'outil fournit :

- Une méthode sécurisée de mise à jour, avec automatisation des copies de sécurité et capacité de rollback
- Un jeu de commandes simples pour faire les mises à jour/réception de livraison
- Une capacité "modulaire" de la livraison contrairement à une livraison compacte

## Prérequis

- 1 serveur sous Linux (n'importe quelle distribution)
- le paquet subversion installé
- les données de proxy subversion configurées (voir la documentation de "subversion derrière un proxy") si votre serveur n'a pas un accès direct à Internet.

## Installer le jeu d'outils

Vous pouvez récupérer le jeu d'outils sur SVN à l'adresse suivante (après avoir eu accès par nos soins)

[http://www.mylearningfactory.com/svn/moodlestore/branches/generic\\_delivery/prodscripts-moodle](http://www.mylearningfactory.com/svn/moodlestore/branches/generic_delivery/prodscripts-moodle)

- Créez un répertoire pour l'installation des outils et checkoutez l'URL ci-dessus dans le répertoire :

```
mkdir /home/prodscripts/monprojet
cd /home/prodscripts/monprojet
svn co
http://www.mylearningfactory.com/svn/moodlestore/branches/generic_delivery/p
rodscripts-moodle
```

## Configurer le jeu d'outils

Le fichier config doit être édité avant l'usage

```
#!/bin/bash
# Real physical dir of the code (absolute path)
REALBASEDIR=/var/www/moodle-%MAJOR%-%HOSTNAME%
# The dir of the real running alias
```

```
BASEDIR=moodle-%MAJOR%- %HOSTNAME%
# The SVN working name
SVNDIR=${BASEDIR}-SVN
# The name for the SAVE copy
SAVEDIR=${BASEDIR}-SAVE
# The name for the SUPERSAVE copy
# The SUPERSAVE copy allows making long term super secured copy
SUPERSAVEDIR=${BASEDIR}-SUPERSAVE
# The name for the CVSD reading user
# You will need to know the remote accepted password for that user
SVNUSER=%SVNUSER%
SVNPASS=%SVNPASS%
# CVS Repository information. Not used for all commands
SVNREPO=%REPO%
SVNROOT=http://www.mylearningfactory.com/svn
MODULE= # obsolete
BRANCH=%BRANCH%
# Apache ownership
FILESERVUSER=<moodleowninguser>
FILESERVGROUP=<moodleowninggroup>
```

Le fichier de configuration travailla par défaut sur des schemas de nommage de type moodle-<major>-<hostname>, par exemple moodle-35-mondomaine pour un moodle en version 3.5. Vous pouvez adopter un autre standard de nommage si vous le souhaitez.

Nous vous avons communiqué :

- %REPO% : la valeur du dépôt que vous utiliserez
- %BRANCH% : la branche de code qui vous correspond
- %SVNUSER% : L'utilisateur du dépôt qui vous est attribué
- %SVNPASS% : le mot de passe associé à cet utilisateur

La configuration des outils prévoit que :

- Moodle soit implanté dans un répertoire quelconque du serveur (suivant la distribution ou les habitudes de l'administrateur système). Ce répertoire **doit être vide** lors du premier checkout.
- Le répertoire des outils contient un lien symbolique vers le répertoire d'installation de Moodle (BASEDIR)
- Le répertoire es outils contient deux répertoires de copie de code de sécurité (SAVEDIR et SUPERSAVEDIR) destinés à stocker des backups de la base de code moodle pendant les livraisons. (il est d'usage de reprendre le nom du lien symbolique suivi respectivement de -SAVE et -SUPERSAVE)
- Le site Moodle opérationnel soit rooté sur le répertoire symbolique, dans le répertoire des outils. Ceci permet au site de continuer à fonctionner sur la copie de sauvegarde pendant les opérations "d'update" à partir du dépôt, et sans que des mises à jour partielles ne viennent déstabiliser le code.

## Commandes



Les mises à jour peuvent être classées dans les cas suivants, en fonction de leur capacité de réversibilité :

- Mise à jour mineure de code sans mutation du modèle de données ou des définitions liées à la version (capacités, événement, modèle de données) ⇒ Réversibilité complète immédiate
- Mise à jour mineure avec mutation locale du modèle de données réversible (la réversibilité suppose que l'ancien code peut opérer le nouveau modèle de données sans faute) ⇒ Réversibilité immédiate
- Mise à jour mineure ou majeure avec modification non réversible u modèle de données ⇒ Réversibilité après réintégration de la sauvegarde de données.

### **./directupdate**

Procède aux mises à jour directes des modification dans la base en exploitation (REALBASEDIR). Les modifications directement descendues ne sont pas intégrées à la copie de sécurité courante.

### **./commit**

Remonte les éventuelles modifications locales de fichiers dans le dépôt SVN. Cette pratique est fortement déconseillée (voir ci-après).

### **./supersyncback**

Réalise une double copie de sécurité dans les répertoires -SAVE et -SUPERSAVE. Ceci peut être utile lors d'une phase de transformation itérative complexe de moodle. (cas rare)

## **Procédures**

Les procédures suivantes permettent d'opérer les opérations courantes de mise en place de code, pourvu que la configuration soit correcte. Le point de départ des commandes est dans le répertoire des outils.

### **Initialisation d'une installation**

```
./checkout      (1)
cp -r moodle-<major>-<hostname>/ .svn .
./syncback     (2)
```

1) Récupération de la base de code complète (la première fois) 2) Création de la copie de sécurité

### **Rechargement complet d'une installation**

```
rm -rf moodle-<major>-<hostname>/ *      (1)
```

```
rm -rf moodle-<major>-<hostname>/..?*      (2)
rm -rf moodle-<major>-<hostname>/.[!.]*    (2)
./checkout
cp -r moodle-<major>-<hostname>/.svn .
./syncback
```

1) Détruit tous les fichiers de l'ancienne base de code 2) Supprime tous les fichiers et répertoires cachés de l'ancienne base de code

### Mise à jour complète

```
./syncback      (1)
./goback        (2)
./update        (3)
./svntoprod     (4)
```

1) Resynchronise la copie de sécurité 2) Bascule l'exploitation sur la copie de sécurité 3) Met à jour la copie d'exploitation relâchée 4) Bascule les modifications en exploitation

### Mise à jour partielle de composants particuliers

```
./syncback
./goback
./update mod/module1
./svntoprod
```

La mise à jour partielle fonctionne sur un container de plugins ou un ensemble de plugins nommés par wildcard, par exemple :

```
./syncback
./goback
./update mod/*
./svntoprod
```

```
./syncback
./goback
./update repo (1)
./svntoprod
```

1) Fera toutes les mises à jour dans les containers report et repository

### Mise à jour d'un fichier racine

```
./syncback
./goback
./update ./index.php
```

```
./svntoprod
```

## Mise à jour rapide d'un composant

```
./pull mod/module1 (1)
```

1) Réalise en une seule commande le cycle de mise à jour partielle sur le composant spécifié.

## Autres manoeuvres

### Remontée de modifications locales

Il est en général déconseillé de modifier les fichiers directement sur le serveur. Mais le cas peut survenir :

- Lors d'un débogage "in situ" sur un site de qualification/recette
- Lors d'un débogage d'urgence sur une situation de production critique.

Les risques principaux de la modification locale des fichiers sur le serveur d'exploitations sont :

- La perte de la modification lors d'une mise à jour ultérieure
- La création d'un conflit SVN qui peut bloquer le fonctionnement complet de la plate-forme (s'il est mal placé)

Néanmoins, si exceptionnellement des modifications locales ont été apportées, il convient de les remonter directement dans le dépôt :

```
./commit mod/module1
```

### Mise à jour directe de la base d'exploitation

Il est possible lors d'une migration importante, de mettre directement en production un ensemble de modifications préalablement qualifiés sur un environnement de pré-recette.

```
./directupdate
```

ou

```
./directupdate mod/module1
```

---

[Guide de démarrage/FAQ sur Moodle](#) - [Catalogue des services](#) - [Aller à l'index des plugins](#) -

A ce jour nous n'avons pas trouvé comment éviter ça

From:

<https://docs.activeprolearn.com/> - **Documentation Moodle ActiveProLearn**

Permanent link:

<https://docs.activeprolearn.com/doku.php?id=delivery:toolkit>

Last update: **2026/01/13 07:19**

