

# Subversion

Pratique des projets collaboratifs avec SVN

**Mike Mason**

Adapté par Isabelle Hurbain

Relecture d'Anne Bougnoux

© Groupe Eyrolles, 2006,  
ISBN : 2-212-11919-4

**EYROLLES**





# Résumé des commandes

---

## Résumé des commandes de Subversion

Récapitulons les commandes de Subversion. Certaines options sont communes à plusieurs d'entre elles ; c'est pourquoi nous les présentons en premier. Lancez `svn help commande` pour savoir avec exactitude quelles options admet telle ou telle commande.

### Options communes

<code>--targets</code>	liste	Passes les éléments de <i>liste</i> en arguments de la commande.
<code>--non-recursive, -N</code>		Ne travailler que sur le répertoire sans chercher à descendre dans les sous-répertoires.
<code>--verbose, -v</code>		Affiche plus d'informations.
<code>--quiet, -q</code>		Affiche le moins d'informations possible.
<code>--username</code>	nom	Indique le nom d'utilisateur à employer pour se connecter et s'authentifier.
<code>--password</code>	mot de passe	Indique le mot de passe à utiliser.

<code>--no-auth-cache</code>		Ne met pas en cache les données d'authentification.
<code>--non-interactive</code>		Ne demande pas d'informations supplémentaires.
<code>--config-dir</code>	répertoire	Lit les options de configuration de l'utilisateur depuis le répertoire indiqué.
<code>--editor-cmd</code>	commande	<i>commande</i> est l'éditeur de messages de journalisation

## svn add

Ajoute les noms des fichiers et des répertoires indiqués au contrôle de version. Ils ne seront effectivement ajoutés que lors de la prochaine propagation.

```
| svn add chemin...
```

### Options

<code>--autoprops</code>	Définit automatiquement les propriétés lors de l'ajout du fichier.
<code>--no-autoprops</code>	Désactive la définition automatique de propriétés.

## svn blame

Alias : `svn ann`, `svn annotate`, `svn praise`.

Affiche les informations concernant la révision et l'auteur de chaque ligne d'un fichier.

```
| svn blame cible ...
```

<code>--revision, -r</code>	<i>révision</i>	Affiche les informations concernant la cible à <i>révision</i> , qui peut aussi bien être une révision unique qu'un intervalle de révisions.
-----------------------------	-----------------	--

## svn cat

Affiche le contenu des fichiers ou adresses indiqué(e)s.

```
| svn cat cible ...
```

### Options

<code>--revision, -r</code>	<i>révision</i>	Affiche le contenu de la cible à la révision indiquée.
-----------------------------	-----------------	--

## svn checkout

Alias : `svn co`.

Extrait une copie de travail du dépôt.

```
| svn checkout adresse ... chemin
```

Extrait les adresses indiquées. Si l'argument *chemin* n'est pas indiqué, l'extraction a lieu dans des répertoires locaux nommés à partir des noms de base des adresses. Si le chemin correspond à un seul argument *adresse*, extrait le dépôt pointé par l'adresse dans le chemin. Si le chemin correspond à plusieurs adresses, extrait les dépôts dans des sous-répertoires nommés d'après les noms de base des adresses.

### Options

<code>--revision, -r</code>	<i>révision</i>	Extrait la révision indiquée.
-----------------------------	-----------------	-------------------------------

## svn cleanup

Nettoie la copie de travail, supprime les verrous, termine les opérations non achevées, etc.

```
| svn cleanup chemin ...
```

## svn commit

Alias : `svn ci`.

Propage au dépôt les modifications que vous avez faites dans votre copie de travail.

```
| svn commit chemin ...
```

### Options

<code>--message, -m</code>	<i>message</i>	Indique le message de journalisation.
<code>--file -F</code>	<i>fichier</i>	Utilise le fichier comme message de journalisation.

## svn copy

Copie un élément dans l'espace de travail ou dans le dépôt et gère l'historique.

```
svn copy src dest
```

*src* et *dest* peuvent être une adresse ou la copie de travail.

<i>src</i>	<i>dest</i>	Action
copie de travail	copie de travail	Copie et planifie l'ajout (y compris de l'historique)
copie de travail	adresse	Propage une copie de l'espace de travail dans l'adresse.
adresse	copie de travail	Extrait l'adresse dans la copie de travail et planifie l'ajout.
adresse	adresse	Copie complètement gérée sur le serveur. Utilisé pour les branches et les marques.

### Options

--revision, -r	<i>révision</i>	La révision de <i>src</i> à copier. N'a de sens que si <i>src</i> est une adresse de dépôt.
----------------	-----------------	---

## svn delete

Alias : `svn del`, `svn remove`, `svn rm`.

Supprime des fichiers et des répertoires du contrôle de versions.

```
svn delete cible ...
```

Supprime les fichiers et les répertoires du dépôt. Si *cible* est une copie de travail ou un répertoire, il est supprimé de la copie de travail et sera supprimé du dépôt lors de la prochaine propagation. Si *cible* est une adresse de dépôt, l'élément est supprimé du dépôt par le biais d'une propagation intermédiaire.

### Options

--message, -m	<i>message</i>	Indique le message de journalisation.
--file -F	<i>fichier</i>	Utilise le fichier comme message de journalisation.

## svn diff

Affiche les différences entre deux chemins.

```
svn diff -rrev1:rev2 cible ...
svn diff adresse1 adresse2
```

Utilisé dans sa première forme, affiche les différences faites à la cible entre les révisions *rev1* et *rev2*. La cible peut être un chemin de l'espace de travail ou une adresse.

Utilisé dans la seconde forme, affiche les différences entre les révisions HEAD des deux adresses.

### Options

--old	<i>argument</i>	<i>argument</i> est la cible la plus ancienne.
--new	<i>argument</i>	<i>argument</i> est la cible la plus récente.

## svn export

Crée une copie non versionnée d'une arborescence.

```
svn export -rrev adresse chemin
```

Exporte vers le répertoire *chemin* une arborescence propre du dépôt indiqué par *adresse*, à la révision *rev* si elle est spécifiée (HEAD dans le cas contraire). Si *chemin* n'est pas indiqué, le dernier composant de l'adresse est utilisé comme nom de répertoire local.

### Options

--revision, -r	révision	Exporte le dépôt à la révision indiquée.
--native-eol	style	Utilise un indicateur de fin de ligne différent de celui indiqué par la propriété <code>svn:eol-style</code> du fichier. <i>style</i> peut prendre les valeurs LF, CR ou CRLF.

## svn import

Propage un fichier ou une arborescence non versionné(e) dans le dépôt.

```
| svn import chemin adresse
```

Propage récursivement une copie de *chemin* vers *adresse*. Si *chemin* est omis, Subversion suppose qu'il s'agit du répertoire courant. Au besoin, les répertoires parents sont créés dans le dépôt.

### Options

--autoprops	Définit automatiquement les propriétés lors de l'ajout du fichier.
--no-autoprops	Désactive la définition automatique de propriétés.

## svn info

Affiche des informations à propos d'un fichier ou d'un répertoire.

```
| svn info chemin ...
```

Affiche les informations correspondant à chaque chemin.

### Options

--recursive, -R	Descend récursivement.
-----------------	------------------------

## svn list

Alias : `svn ls`.

Fait la liste de tous les éléments du dépôt.

```
| svn list cible ...
```

Fait la liste de tous les fichiers et répertoires de *cible* tels qu'ils sont actuellement présents dans le dépôt. Si *cible* est le chemin d'une copie de travail, l'adresse correspondante est utilisée.

### Options

--verbose, -v	Affiche des informations supplémentaires pour chaque répertoire.
---------------	--

## svn log

Affiche les messages de journalisation d'un ensemble de révisions ou de fichiers.

```
svn log cible
```

Affiche les messages de journalisation d'un chemin local ou d'une adresse de dépôt. Pour un chemin local, l'intervalle de révision par défaut est BASE:1 et pour une adresse, l'intervalle de révision par défaut est HEAD:1.

### Options

<code>--revision, -r</code>	<i>révision</i>	Si <i>révision</i> est une révision unique, affiche le message de journalisation de cette révision. Si c'est un intervalle de révisions, affiche tous les messages correspondant à cet intervalle.
<code>--verbose, -v</code>		Affiche les chemins concernés dans chaque message de journalisation.
<code>--stop-on-copy</code>		Ne dépasse pas les copies lors du parcours de l'historique. Utile pour connaître le point de départ d'une branche.

## svn merge

Fusionne les différences entre deux codes sources dans une copie de travail.

```
svn merge adresse1@rev1 adresse2@rev2 chemin
svn merge chemin1@rev1 chemin2@rev2 chemin
svn merge -r rev1:rev2 source chemin
```

Utilisé sous la première forme, compare les adresses aux révisions respectives *rev1* et *rev2* (ou HEAD si les numéros de révision ne sont pas indiqués).

Utilisé sous la seconde forme, compare les dépôts correspondants aux copies de travail indiquées. Les numéros de révision sont obligatoires.

Utilisé sous la troisième forme, *source* peut être une adresse ou un élément de la copie de travail, auquel cas l'adresse correspondante est utilisée. Cette adresse est comparée entre ses révisions *rev1* et *rev2*.

*chemin* est la copie de travail dans laquelle seront enregistrées les modifications. Si *chemin* est omis, Subversion utilise le répertoire courant, à moins que les deux adresses n'aient le même nom de base, correspondant à un fichier du répertoire, auquel cas les différences seront appliquées à ce fichier.



## Options

<code>--diff3-cmd</code>	<i>cmd</i>	Utilise <i>cmd</i> comme commande pour la fusion.
<code>--ignore-ancestry</code>		Ignore l'ancienneté lors du calcul de la fusion.

## svn mkdir

Crée un nouveau répertoire dans le contrôle de versions.

```
| svn mkdir cible ...
```

Les répertoires indiqués par un chemin local sont créés localement et ne seront effectivement ajoutés au dépôt que lors de la prochaine propagation. Les répertoires indiqués comme une adresse sont créés dans le dépôt par le biais d'une propagation intermédiaire.

Dans les deux cas, les répertoires parents doivent être pré-existants.

## svn move

Alias : `svn mv`, `svn rename`, `svn ren`.

Déplace ou renomme un élément dans la copie de travail ou dans le dépôt.

```
| svn move src dest
```

*src* et *dest* doivent être tous deux des chemins de la copie de travail ou tous deux des adresses de dépôt. Dans la copie de travail, l'élément est déplacé et le nouvel emplacement sera ajouté à la prochaine propagation. Dans le dépôt, le déplacement est fait entièrement et immédiatement du côté du serveur.

## Options

<code>--revision, -r</code>	<i>révision</i>	Déplace la révision indiquée.
-----------------------------	-----------------	-------------------------------

## svn propdel

Alias : `svn pdel`, `svn pd`.

Supprime une propriété d'un fichier ou d'un répertoire.

```
| svn propdel nom chemin ...
```

Supprime la propriété *nom chemin* dans la copie de travail locale.

## svn propedit

Alias : `svn pedit`, `svn pe`.

Modifie une propriété sur un fichier ou un répertoire.

```
| svn propedit nom chemin ...
```

Lance l'éditeur externe pour modifier la propriété *nom chemin* dans la copie de travail locale.

## svn propget

Alias : `svn pget`, `svn pg`.

Affiche la valeur d'une propriété d'un fichier ou d'un répertoire.

```
| svn propget nom chemin ...
```

Affiche le contenu de la propriété *nom* de chaque élément de *chemin*. Par défaut, Subversion ajoute un saut de ligne à la fin des valeurs de la propriété pour rendre la sortie présentable. Quand plusieurs chemins sont indiqués, chaque valeur de propriété est préfixée par le chemin auquel elle est associée.

### Options

<code>--strict</code>	Supprime les sauts de ligne supplémentaires et autres éléments de formatage. Utile lorsqu'il s'agit de rediriger des propriétés binaires vers un fichier.
-----------------------	---

## svn proplist

Alias : `svn plist`, `svn pl`.

Fait la liste de toutes les propriétés sur les fichiers ou répertoires *chemin*.

```
| svn proplist chemin ...
```

### Options

<code>--verbose, -v</code>		Affiche des informations supplémentaires.
<code>--recursive, -R</code>		Descend récursivement dans les sous-répertoires.
<code>--revision, -r</code>	<i>révision</i>	Fait la liste des propriétés définies dans la révision indiquée du <i>chemin</i> .

## svn propset

Alias : svn pset, svn ps.

Définit une propriété sur des fichiers ou des répertoires.

```
svn propset nom valeur chemin ...
```

Définit la propriété *nom* à la valeur *valeur* sur *chemin*. Si *valeur* n'est pas définie, vous devez utiliser l'option -F pour indiquer un fichier dont le contenu sera la valeur de la propriété.

### Options

--file, -F	<i>fichier</i>	Lit le contenu du fichier et l'utilise comme valeur de la propriété.
--recursive, -R		Descend récursivement dans les sous-répertoires.
--encoding	<i>encodage</i>	Définit la valeur dans l'encodage spécifié.

## svn resolved

Supprime l'état de conflit sur des fichiers ou des répertoires de la copie de travail.

```
svn resolved chemin ...
```

Lève l'état de conflit d'un fichier. Cette commande ne résout pas sémantiquement un conflit et n'en supprime pas les marques ; elle se contente de supprimer les fichiers artefacts liés au conflit et permet de propager le fichier *chemin*.

### Options

--recursive, -R	Descend récursivement dans les sous-répertoires.
-----------------	--

## svn revert

Restaure le fichier initial de la copie de travail (supprime la plupart des modifications locales).

```
svn revert chemin ...
```

Cette commande ne nécessite pas d'accès réseau et annule la majorité des modifications faites à *chemin*. Elle ne rétablit cependant pas les répertoires supprimés.

## Options

<code>--recursive, -R</code>	Descend récursivement dans les sous-répertoires.
------------------------------	--

## svn status

Alias : `svn stat`, `svn st`.

```
| svn status chemin ...
```

Sans argument, affiche les éléments modifiés localement (ne nécessite pas d'accès au réseau). L'option `-u` ajoute les révisions de travail et les informations obsolètes par rapport au serveur. L'option `-v` affiche les informations complètes des révisions de chaque élément.

Les cinq premières colonnes sont chacune larges d'un caractère.

La première colonne indique si un élément a été ajouté, supprimé ou modifié.

	Pas de modification.
A	Ajouté.
C	En conflit.
D	Supprimé.
G	Fusionné.
I	Ignoré.
M	Modifié.
R	Remplacé.
X	Élément non versionné mais utilisé par une définition externe.
?	Élément non versionné.
!	Élément manquant (supprimé par une commande non svn) ou incomplet
~	Élément versionné caché par un élément d'un autre type.

La seconde colonne affiche les modifications des propriétés.

	Pas de modification.
C	En conflit.
M	Modifié.

La troisième colonne indique si le fichier ou le répertoire est verrouillé.

	Non verrouillé.
V	Verrouillé.

La quatrième colonne indique si la propagation contiendra des informations d'historique.

	Pas d'historique.
+	Propagation avec historique.

La cinquième colonne signale si un élément a changé de dépôt par rapport à son parent.

	Normal.
S	Passé dans un autre dépôt.

Les informations d'obsolescence apparaissent dans la huitième colonne lorsque l'option `-u` est passée.

*	Une révision plus récente existe sur le serveur
	La copie de travail est à jour.

Les champs restants sont de taille variable et délimités par des espaces : il s'agit de la révision de travail (lorsque `-u` ou `-v` est spécifié), de la dernière révision propagée et de l'auteur de cette dernière propagation (avec `-v`). La copie de travail est toujours le dernier champ et peut donc inclure des espaces.

## Options

<code>--show-updates, -u</code>	Contacte le serveur et affiche les informations de mise à jour.
<code>--verbose, -v</code>	Affiche des informations supplémentaires.
<code>--non-recursive, -N</code>	Ne travaille que sur un seul répertoire.
<code>--no-ignore</code>	Ne prend pas en compte les fichiers ignorés par défaut ou par une propriété <code>svn:ignore</code> .

## svn switch

Alias : `svn sw`.

Associe la copie de travail à une adresse différente.

```
svn switch adresse chemin
```

Met à jour la copie de travail pour qu'elle reflète une nouvelle adresse du dépôt. Ce comportement est similaire à celui de `svn update` et permet d'associer une copie de travail à une branche ou une marque en restant dans le même répertoire.

### Options

<code>--revision, -r</code>	<i>révision</i>	Passer à la révision indiquée
<code>--non-recursive, -N</code>		Ne travaille que sur un seul répertoire.
<code>--diff3-cmd</code>	<i>cmd</i>	<i>cmd</i> est la commande de fusion.

## svn update

Alias : `svn up`.

Met à jour la copie de travail par rapport au dépôt.

```
svn update path ...
```

Si aucune révision n'est indiquée, met à jour la copie de travail avec HEAD. Sinon, synchronise la copie de travail avec la révision indiquée.

Pour chaque élément mis à jour, une ligne commençant par un des caractères suivants indique l'action qui a été menée :

A	Ajouté.
D	Supprimé.
U	Actualisé.
C	En conflit.
M	Fusionné.

La première colonne indique l'état du fichier lui-même, la seconde colonne indique l'état des propriétés de ce fichier.

## Options

<code>--revision, -r</code>	<i>révision</i>	Passer à la révision indiquée
<code>--non-recursive, -N</code>		Ne travailler que sur un seul répertoire.
<code>--diff3-cmd</code>	<i>cmd</i>	<i>cmd</i> est la commande de fusion.

## Comment faire pour...

### Extraire (chapitre 3)

```
| svn checkout adresse chemin
```

### Extraire une révision donnée (chapitre 4)

```
| svn checkout -r revision adresse
```

### Extraire la révision d'une date donnée (chapitre 4)

```
| svn checkout -r « {date} » adresse
```

### Savoir d'où vient une copie de travail (chapitre 4)

```
| svn info chemin
```

### Actualiser une copie de travail (chapitre 4)

```
| svn update
```

### Actualiser des éléments donnés d'une copie de travail (chapitre 4)

```
| svn update chemin ...
```

### Ajouter des fichiers à un dépôt (chapitre 4)

```
| svn add chemin ...
```

**Définir une propriété sur un fichier ou un répertoire (chapitre 6)**

```
svn propset nom valeur chemin ...
```

**Modifier une propriété sur un fichier ou sur un répertoire (chapitre 6)**

```
svn propedit nom chemin ...
```

**Faire la liste de toutes les propriétés sur un fichier ou sur un répertoire (chapitre 6)**

```
svn proplist chemin ...
```

**Afficher le contenu d'une propriété (chapitre 6)**

```
svn propget nom chemin ...
```

**Supprimer une propriété (chapitre 6)**

```
svn propdel nom chemin ...
```

**Activer l'expansion de mots-clés pour un fichier (chapitre 6)**

```
svn propset svn:keywords « mots-clés » fichier ...
```

**Ignorer certains fichiers dans un répertoire (chapitre 6)**

```
svn propedit svn:ignore chemin ...
```

**Définir le caractère de fin de ligne d'un fichier (chapitre 6)**

```
svn propedit svn:eol-style style chemin ...
```

**Définir le type MIME d'un fichier (chapitre 6)**

```
svn propset svn:mime-type type-mime chemin ...
```



**Marquer un fichier comme exécutable (chapitre 6)**

```
| svn propset svn:executable true chemin ...
```

**Copier un fichier ou un répertoire (chapitre 4)**

```
| svn copy source destination
```

**Renommer un fichier ou un répertoire (chapitre 4)**

```
| svn rename ancien nom nouveau nom
```

**Déplacer un fichier ou un répertoire (chapitre 4)**

```
| svn move source destination
```

**Afficher les changements dans un fichier ou un répertoire (chapitre 4)**

```
| svn diff chemin
```

**Comparer deux révisions d'un fichier (chapitre 4)**

```
| svn diff -rrev1:rev2 fichier
```

**Afficher les différences entre un fichier et la dernière révision du dépôt (chapitre 4)**

```
| svn diff -r HEAD fichier ...
```

**Afficher les modifications les plus récentes d'un fichier (chapitre 4)**

```
| svn diff -r PREV:BASE fichier ...
```

**Créer un fichier patch (chapitre 4)**

```
| svn diff > fichier
```

**Appliquer un patch (chapitre 4)**

```
| patch -p0 -i fichier
```

**Supprimer vos modifications face à un conflit (chapitre 5)**

```
| svn revert fichier ...  
| svn update fichier ...
```

**Supprimer les modifications d'une autre personne face à un conflit (chapitre 5)**

```
| cp fichier.mine fichier  
| svn resolved fichier
```

**Marquer un conflit comme résolu (chapitre 5)**

```
| svn resolved fichier
```

**Propager des modifications (chapitre 4)**

```
| svn commit -m « message »
```

**Afficher l'historique d'un fichier (chapitre 4)**

```
| svn log fichier
```

**Afficher l'activité récente d'un répertoire (chapitre 4)**

```
| svn log chemin | more
```

**Afficher l'historique détaillé d'un fichier (chapitre 4)**

```
| svn log -v fichier ...
```

**Annoter un fichier avec les informations sur ses auteurs (chapitre 4)**

```
| svn blame fichier ...
```

**Poser un verrou sur un fichier (chapitre 5)**

```
| svn lock fichier
```

**Supprimer un verrou en tant qu'administrateur (chapitre 5)**

```
| svnadmin rmllocks depot fichier
```

**Supprimer un verrou en tant qu'utilisateur (chapitre 5)**

```
| svn unlock --force fichier
```

**Voler un verrou (chapitre 5)**

```
| svn lock --force fichier
```

**Revenir sur une modification déjà propagée (chapitre 5)**

```
| svn merge -r rev:rev-1 chemin ...
```

**Vérifier l'état de la copie de travail (chapitre 4)**

```
| svn status
```

**Afficher les mises à jour en attente par rapport au dépôt (chapitre 4)**

```
| svn status --show-updates
```

**Créer une branche de livraison (chapitre 8)**

```
| svn copy \  
| svn://monserveur/projet/trunk \  
| svn://monserveur/projet/branches/BL-x.y
```

**Extraire une branche de livraison (chapitre 8)**

```
| cd travail  
| svn checkout svn://monserveur/projet/branches/BL-x.y
```

### Associer une copie de travail à une branche de livraison (chapitre 8)

```
cd monprojet
svn switch svn://monserveur/projet/branches/BL-x.y
```

### Associer une copie de travail au tronc commun (chapitre 8)

```
cd monprojet
svn switch svn://monserveur/projet/trunk
```

### Créer une marque de livraison (chapitre 8)

```
svn copy \
  svn://monserveur/projet/branches/BL-x.y \
  svn://monserveur/projet/marques/LIV-x.y
```

### Extraire une livraison (chapitre 8)

```
svn checkout svn://monserveur/projet/tags/LIV-x.y
```

### Fusionner un correctif de bogue simple depuis une branche de livraison vers le tronc commun (chapitre 8)

```
cd projet
svn update
svn merge -rrev-1:rev svn://myserver/projet/branches/BL-x.y
```

### Créer une branche pour un correctif complexe (chapitre 8)

```
svn copy \
  svn://monserveur/projet/branches/BL-x.y \
  svn://monserveur/projet/branches/BUG-numéro
svn copy \
  svn://monserveur/projet/branches/BUG-numéro \
  svn://monserveur/projet/marques/PRE-numéro
```

### Extraire une branche de correctif de bogue (chapitre 8)

```
svn checkout svn://monserveur/projet/branches/BUG-numéro
```

### Marquer la fin d'un correctif (chapitre 8)

```
svn copy \  
  svn://monserveur/projet/branches/BUG-numéro \  
  svn://monserveur/projet/marques/POST-numéro
```

### Fusionner un correctif de bogue complexe dans une branche de livraison (chapitre 8)

```
cd BL-x.y  
svn merge \  
  svn://monserveur/projet/marques/PRE-numéro  
  svn://monserveur/projet/marques/POST-numéro
```

### Créer des branches expérimentales (chapitre 8)

```
svn copy \  
  svn://.../trunk  
  svn://.../branches/EXP-initiales-description
```

### Utiliser une branche expérimentale (chapitre 8)

```
svn switch svn://.../branches/EXP-initiales-description
```

### Revenir au tronc commun (chapitre 8)

```
svn switch svn://.../trunk
```

### Savoir quand a été créée une branche (chapitre 8)

```
svn log --stop-on-copy  
  svn://.../branches/branche
```

### Fusionner une branche expérimentale (chapitre 8)

```
svn log --stop-on-copy \  
  svn://.../branches/EXP-initiales-description  
cd travail  
svn merge \  
  -r revision-début-branche:HEAD\  
  svn://.../branches/EXP-initiales-description  
svn commit
```

## Importer un projet dans le dépôt (chapitre 9)

```
cd projet
svn import svn://monserveur/projet/trunk
```

## Créer des répertoires pour un projet (chapitre 9)

```
svn mkdir svn://monserveur/projet/
svn mkdir svn://monserveur/projet/trunk
svn mkdir svn://monserveur/projet/tags
svn mkdir svn://monserveur/projet/branches
```

## Importer du code de tierce partie (chapitre 10)

```
svn import arborescence-fournisseur \
  svn://.../vendormsg/fournisseur/produit/current
```

## Marquer un paquetage fournisseur (chapitre 10)

```
svn copy \
  svn://.../vendormsg/fournisseur/produit/current \
  svn://.../vendormsg/fournisseur/produit/version
```

## Charger un paquetage fournisseur (chapitre 10)

```
svn_load_dirs.pl \
  svn://.../vendormsg/fournisseur/produit\
  current arborescence-fournisseur
```

## Utiliser du code d'un fournisseur dans un projet (chapitre 10)

```
svn copy \
  svn://.../vendormsg/fournisseur/produit/version
svn://.../project/trunk/vendor/produit
```

## Mettre à jour le code d'un fournisseur dans un projet (chapitre 10)

```
svn merge \
  svn://.../vendormsg/fournisseur/produit/ancien \
  svn://.../vendormsg/fournisseur/produit/nouveau \
  vendor/produit
```

**Démarrer svnservice sous Windows (chapitre 11)**

```
| start svnservice --daemon --root rep-depot
```

**Démarrer svnservice sous Unix (chapitre 11)**

```
| svnservice --daemon --root rep-depot
```

**Créer une sauvegarde complète de votre dépôt (chapitre 11)**

```
| svnadmin dump depot > image
```

**Créer une sauvegarde incrémentale de votre dépôt (chapitre 11)**

```
| svnadmin dump --incremental --revision rev1:rev2 depot
```