DÉPARTEMENT
Sciences Informatiques

Licence 2 I
2012–2013
Génie Logiciel
T.T. Dang Ngoc
dntt@u-cergy.fr

TD2 - Outils de développement : SVN

L'objectif de ce TD est de se familiariser avec SVN et d'en entrevoir les possiblités. L'exercice consistera à enregistrer le projet GENLOG sous SVN, puis de faire un certain nombre de manipulations (ajout, modification, gestion de conflits).

Ce TP se fait **en binôme**, chacun ayant une machine. Dans la suite de l'exercice, on désignera par (pierre) et (marie) les identificateurs (login) respectifs des deux étudiants du binôme. Le répertoire SVN (qu'on appelle `repository`) est placé sur une machine externe appelée `depinfo.u-cergy.fr` dans un répertoire appelé `/usvn/svn/`*`votre_nom_de_module`* Il faut bien noter que sur un système réel de production, ce répertoire se trouve normalement dans un répertoire indépendant, voire (c'est le cas le plus fréquent) sur une machine différente.

Dans la suite, (pierre) (resp. (marie)) signifiera que l'opération ne doit être réalisée que par pierre (resp. marie), (1/2) par seulement un des deux, et (2/2) , par les deux.

1 Importation du projet

(marie) Récupérez le fichier `genlog.tgz` et désarchivez le en tapant : `tar xzfv genlog.tgz` Vous obtiendrez un répertoire PROJET-GENLOG.

(marie) Allez dans le répertoire PROJET-GENLOG, puis importez le en tapant :

```
svn import -m GENLOG http://depinfo.u-cergy.fr/usvn/svn/nom_module/GENLOG
```

Le projet est à présent enregistré dans le repository sous le nom GENLOG.

2 Récupération du projet

Chacune des personnes va récupérer une version du projet pour ensuite travailler chacun de son côté. Pour cela :

- (2/2) exportez le projet en tapant :
  `svn checkout  http://depinfo.u-cergy.fr/usvn/svn/nom_module`
- (marie) vérifiez avant toute modification que le projet ainsi exporté est identique au répertoire PROJET-GENLOG exporté à l'exception des répertoires SVN/ qui ont été créés. Pour cela tapez
  `diff -r PROJET-GENLOG GENLOG`
- (marie) le répertoire PROJET-GENLOG ne vous sert désormais plus, vous pouvez l'effacer si vous voulez.

Chez chacun d'entre vous, le répertoire de travail est à présent GENLOG.

3 Modifications

Dans cette section, vous allez apprendre à modifier, supprimer et ajouter des fichiers chacun de votre côté tout en se synchronisant de temps en temps. Pour cela, vous effectuerez les opérations suivantes.

Pour les tâches qui vont suivre, (pierre) va s'occuper de tout ce qui concerne les chiens, et (marie) de tout ce qui concerne les oiseaux. Et pour les poissons, c'est le premier qui s'y colle. Ne vous concertez pas (pensez que vous pouvez travailler en différé à des endroits différents).

Dans le répertoire `GENLOG/src/gl/zoo` :
- (pierre) Modifiez la classe `PitBull` pour ajouter la méthode `mordre`.
- (marie) Créer les classes `Canari` héritant de `Oiseau`.
- (pierre) Créer la classe `Labrador` héritant de `Chien` et qui a les méthodes `nager`, `rapporterBaton`, et `jouer`.
- (marie) Créer les classes `Perroquet` héritant de `Oiseau` et qui a la méthode `parler`.
- (1/2) ou (2/2) Créer la classe `Carpe` héritant de `Poisson`.
- (pierre) Supprimer la classe `Caniche`.
- (2/2) Créer d'autres classes et/ou méthodes dans le `zoo`.
- (2/2) **Au fur et à mesure**, modifiez la classe `TestZoo` pour tester vos nouvelles méthodes/classe

Evidemment, au fur et à mesure que vous écrivez les méthodes et répertoires, vous faites de temps en temps un `commit`. En tapant

```
svn commit truc
```

où `truc` représente un nom de fichier, un nom de répertoire, ou rien (ce qui veut dire le répertoire courant). Avant de commiter, tester toujours vos classes en compilant `TestZoo`. Avant l'écriture d'une nouvelle classe/méthode, il peut être utile de faire une mise à jour

```
svn update truc
```

pour intégrer les nouvelles modifications de votre binôme.

Si vous expérimentez des conflits durant votre développement, résolvez-les. Si vous n'en rencontrez pas, créez en un en vous concertant avec votre binôme en décidant de modifier quelque chose au même endroit.

Utilisez `svn diff -r` pour voir les différences entre deux versions.

Regardez régulièrement l'état de vos versions, avec `svn status`, `svn history` et `svn annotate` (voir les autres commandes disponibles avec `svn help`.

4 Tags

Au lieu d'utiliser les numéros d'enregistrement (qui sont individuels pour chache fichier géré par SVN), il est possible d'étiqueter un ensemble de données avec un nom de version commun, et d'utiliser ensuite cette étiquette pour récuperer les données.

Une fois les modifications demandées dans l'exercice précédent effectuées et que tout fonctionne, étiquetez votre version courante avec le tag `V2009-VERSION-COMMERCIALISE`, puis committez.

```
svn copy nom_du_tag fichiers
```

Ensuite refaites quelques modifications (avec un `commit`), puis récuperez la version du tag.

5 Branchements

La commande tag permet aussi de faire des branchements.

```
svn copy -r tagname fichiers
```

Revenez à une version antérieure de votre projet et créez une nouvelle branche, avec des modifications nouvelles que vous enregistrerez avec `commit`. Constatez avec `status` que les numéros de séquence ont été allongés.