

Installation de JabRef en environnement LINUX et / ou FreeBSD

Jean-Marc LICHTLE

14 octobre 2004

Table des matières

1 Introduction	1
2 Installation avec LINUX	2
2.1 Récupération du sdk de Java	2
2.2 Téléchargement de JabRef.jar	2
2.2.1 Premiers essais	3
3 Installation sous FreeBSD	3
3.1 Installation "crade"	3
3.1.1 Récupération des binaires compilés pour LINUX	3
3.1.2 Récupération de JabRef	3
3.1.3 Premiers essais	3
3.2 Installation au moyen des ports	4
3.2.1 Mise à jour des ports	4
3.2.2 Installation en utilisant le port mis à jour	4
3.2.3 Une dernière retouche	5
4 Conclusion	5

En deux mots :

Ce document a pour vocation de montrer comment installer JabRef, un logiciel de gestion de bibliographie en environnement LINUX ou FreeBSD. Dans une version future je complèterais peut-être ce document en faisant une description du fonctionnement de JabRef.

1 Introduction

Pourquoi diable s'intéresser à JabRef? La réponse est simple, la gestion de données bibliographiques est un point important dans certaines situations, notamment lorsqu'on effectue des travaux de recherche. Plus généralement toutes les données ordonnées qui sont susceptibles d'être référencées dans un document de synthèse type rapport, thèse, mémoire etc. peuvent être efficacement traitées par un gestionnaire de données bibliographiques.

Il existe, dans le monde des ordinateurs tournant sous un système d'exploitation créé du côté de REDMOND (USA) bon nombre de logiciels convenant à cette tâche. Nous nous intéresserons ici toutefois à JabRef parce qu'il présente l'avantage de tourner avec le système de Bill mais aussi avec les systèmes libres, notamment LINUX et FreeBSD.

Ce document ne détaille que l'installation sur des systèmes libres, pour les systèmes commerciaux veuillez appeler les hot lines mises à votre disposition.

2 Installation avec LINUX

Cette description est relative à une installation sur LINUX Mandrake 10.0. Elle est certainement applicable directement à bon nombre d'autres distributions, du moins celles qui utilisent rpm.

2.1 Récupération du sdk de Java

L'abréviation sdk signifie Software Development Kit. Ce kit est nécessaire au fonctionnement de JabRef, c'est donc ici que nous allons commencer notre travail.

Commencez par télécharger le sdk en vous rendant à l'adresse

`http://sun.java.com/j2se/1.4.2/download.html`.

Suivez le lien Download J2SE SDK situé dans le milieu de la page, acceptez la licence qui vous est proposée, après quoi vous arrivez à la page de chargement. Téléchargez maintenant le fichier `j2sdk-1_4_2_05-linux-i586-rpm.bin`. Je pars du principe que vous disposez d'une ligne ADSL faute de quoi vous avez le temps ici d'aller tondre la pelouse, le bébé pèse en effet 33.60 MB! Rendez ce binaire exécutable avec `chmod +x j2sdk...` puis lancer le. Ce faisant vous allez générer un fichier rpm. La génération n'est pas directe puisque, à nouveau, on vous demande d'accepter la licence et tapant explicitement `yes` après avoir défilé toute la licence au moyen de la barre espace.

Il reste maintenant à installer le rpm, action qui nécessite d'être root et utilise la syntaxe :

```
# rpm -hiv j2sdk-1_4_2_05-linux-i586-rpm
```

A ce stade vous vérifiez si tout c'est bien passé en tapant simplement :

```
# which java
```

La commande `which` recherche une commande dans tous les répertoires figurant dans le PATH et vous indique en sortie le chemin trouvé.

Si vous obtenez une réponse du style :

```
which: no java in (/usr/bin:/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin/  
:/usr/games:/home/jml/bin)
```

c'est que le binaire nommé `java` a été installé ailleurs de dans les sous répertoires désignés comme chemin d'accès. En fait, et vous le vérifierez aisément, le binaire a été installé dans `/usr/java/j2sdk.../bin/`. Surtout ne touchez à rien, n'essayez pas de le déplacer ou d'ajouter ce sous répertoire dans le PATH, le plus simple est de placer, depuis un sous répertoire désigné dans PATH, un lien symbolique vers `java` ce qui se fait au moyen de la commande

```
# ln -s /usr/java/j2sdk1.4.../bin/java /usr/bin/java
```

Cette ligne de commande crée une nouvelle entrée dans `/usr/bin`, entrée qui est un lien vers l'exécutable `java` comme l'indique la lettre `l` qui débute la ligne si vous faite un `ls -alrt java`.

2.2 Téléchargement de JabRef.jar

Téléchargez maintenant `JabRef.jar` en vous rendant sur

`http://sourceforge.net/projects/jabref/`

et en suivant le lien `Download`

2.2.1 Premiers essais

Il suffit maintenant de taper `java -jar Jabref-1.5.jar` pour voir se lancer l'application tant souhaitée. Attention, cette application ne se lance que depuis le répertoire où elle a été chargée !

3 Installation sous FreeBSD

FreeBSD est capable, moyennant une installation *ad_hoc*, d'exécuter tous les binaires compilés pour LINUX. Cette possibilité va être mise à profit ici. Nous allons en fait reprendre le schéma ci-dessus en le modifiant très peu pour l'adapter à FreeBSD.

3.1 Installation "crade"

Le titre est très évocateur, puriste passe ton chemin, saute au chapitre suivant, ici c'est fait pour les dégoûtants, plus précisément les utilisateurs qui, ne faisant pas une utilisation intensive de la gestion bibliographique, ont simplement besoin de faire fonctionner le temps de faire un travail précis.

3.1.1 Récupération des binaires compilés pour LINUX

Suivez le schéma ci-dessus mais cette fois, au lieu de récupérer le fichier rpm, récupérez le fichier bin dont le lien se situe une ligne en dessous du précédent. Le poids du bébé est le même, vous avez le temps de vous faire un café. Rendre ce fichier exécutable puis l'exécuter crée instantanément un sous répertoire partant du répertoire courant dans lequel on vient de télécharger le binaire, sous répertoire nommé `j2sdk1.4.2_05/`. L'exécutable java donc nous avons besoin pour faire tourner notre application se trouve dans `j2sdk1.4.2_05/jre/bin/`.

3.1.2 Récupération de JabRef

Voir le chapitre consacré à l'installation sous LINUX.

3.1.3 Premiers essais

La syntaxe est très proche de celle utilisée sous LINUX si ce n'est que l'exécutable java n'est pas dans le PATH ce qui oblige à préciser sa localisation en clair :

```
$ j2sdk1.4.2_05/jre/bin/java -jar JabRef.jar
```

Il va sans dire que l'établissement d'un lien symbolique peut être ici d'un grand secours pour simplifier la syntaxe et la ramener à celle qui était employée sous LINUX. Pour établir ce lien tapez simplement :

```
$ ln -s j2sdk1.4.2_05/jre/bin/java java
```

après quoi il vous suffira de taper

```
./java -jar Jabref-1.5.jar
```

pour lancer l'application. Notez la présence cette fois de `./` devant le nom de la commande. Ce détail est nécessaire parce que le répertoire dans lequel vous avez, à priori installé le lien, en principe votre répertoire personnel, n'est pas désigné dans le chemin de recherche.

3.2 Installation au moyen des ports

Les ports, si vous les avez mis en place lors de l'installation de FreeBSD sont d'innombrables sous répertoires, un par application, contenant chacun quelques fichiers, certains décrivant le logiciel visé par le sous-répertoire, d'autre concentrant, dans un makefile, la méthode de récupération des sources, de compilation de celles-ci et d'installation. L'utilisateur intéressé se reportera utilement à la littérature portant sur le sujet sur différents sites Internet.

La théorie voudrait qu'un "make install clean" règle le problème en deux coups de cuillère à pot. Il n'en est malheureusement rien. Le sdk n'étant pas un logiciel open source le make s'arrête très rapidement en indiquant la nécessité de récupérer le binaire et de le copier à la main dans /usr/ports/distfiles, le sous répertoire dans lequel sont copiés par défaut tous les paquets logiciels téléchargés en appliquant le système des ports. Cette difficulté n'est malheureusement pas unique. En effet, à l'heure où j'écris ces lignes (octobre 2004) de l'eau a coulé sous les ponts depuis que ma version 5.2.1 de FreeBSD a été assemblée. Il en résulte que le makefile d'origine est en retard de 3 versions sur le fichier mis à disposition par SUN. Qu'à cela ne tienne allez vous me dire, renommons le fichier pour lui donner le nom de la version désignée dans le makefile ou bricolons ce dernier pour changer le numéro de version ! Ne cherchez pas, j'ai essayé. Des clous ! Dans le raisonnement nous n'avons pas tenu compte de la vérification systématique de la somme de contrôle ! Sans appel, les make successifs échouent systématiquement. Il faut donc bien se rendre à l'évidence, il est nécessaire de passer par une mise à jour des ports, en priant le ciel que le nouveau port prenne bien en compte précisément la version téléchargée !

3.2.1 Mise à jour des ports

La première fois on a un peu les mains moites mais finalement ça se passe très bien. Sans entrer dans le détail des explications les opérations à faire sont les suivantes :

- Installer le port de cvsup, le logiciel qui va nous permettre dans un instant de faire la mise à jour. Pour ce faire RV dans le sous répertoire /usr/ports/net/cvsup-without-gui, tapez make, puis une fois la compilation terminée make install et enfin make clean. Dans le monde FreeBSD on passe l'aspirateur systématiquement quand on a fini de travailler.... Une autre variante est de taper directement make install clean pour tout grouper sur une seule ligne de commande.
- Maintenant que cvsup est en place copiez le fichier /usr/share/examples/cvsup/port-supfile dans /etc puis éditez la copie pour remplacer la ligne default host=CHANGE_THIS.FreeBSD.org par default host=cvsup.fr.freebsd.org.
- Il suffit maintenant de lancer cvsup en lui donnant l'adresse du bon fichier de paramètres en tapant
`cvsup /etc/ports-supfile`
et vous voilà parti pour un long moment à regarder défiler devant vos yeux hilares la liste de toutes les tâches accomplies lors de la mise à jour. Si la pelouse n'est pas trop énorme et si le temps s'y prête c'est le moment de se demander où est rangée la tondeuse.

3.2.2 Installation en utilisant le port mis à jour

Début octobre 2005 la combinaison est idéale puisque le port mis à jour désigne très exactement la version de j2sdk qui est téléchargeable sur le site de SUN. En soi cela n'a rien d'étonnant, la mise à jour sert précisément à ça, mais il se pourrait que le hasard maléfique vous fasse faire l'opération décrite plus haut précisément après un changement de version et avant la mise à jour du port. Dans ce cas il faudrait peut-être attendre un jour ou deux avant de refaire un cvsup. Dans les bonnes conditions décrites la mise en place du j2sdk est un vrai régal, make, make install, make clean ou plus simplement make install clean.

3.2.3 Une dernière retouche

Arrivé à ce stade vous constatez de java ne fonctionne toujours pas. Oui, mais si vous avez bien suivi vous vous dites certainement, "il manque un lien, java est installé hors de portée du PATH" Et oui. En cherchant un peu vous trouvez rapidement le fichier java dans /usr/local/linux-sun-jdk1.4.2/jre/bin. Il suffit donc de faire un lien comme suit :

```
# ln -s /usr/local/linux-sun-jdk1.4.2/jre/bin/java /usr/bin/java
```

pour que tout rentre dans l'ordre puisque, évidemment /usr/bin est visité à chaque fois que l'interpréteur de commande cherche un exécutable.

Voilà, si vous aviez utilisé la première méthode c'est le moment de faire marche arrière en effaçant tout et en vous disant fièrement que s'était juste pour essayer, qu'il n'était pas question que sur votre machine il y ait de façon durable une installation aussi dégoûtante.

4 Conclusion

Cette petite excursion du côté de JabRef nous aura permis de traiter le même problème sous deux systèmes d'exploitation différents. J'aime bien cette dualité d'approche qui offre un éclairage réciproque et permet d'ouvrir un peu le sujet au delà de la simple explication de l'utilisation de la commande rpm. L'exposé développé dans la première partie est partiellement applicable à la seconde, un bon exemple est la création d'un lien symbolique, son utilité, sa syntaxe. La seconde partie nous a permis d'aborder, d'effleurer plus exactement deux sujets extrêmement intéressants, l'utilisation de binaires compilés pour LINUX par FreeBSD et la technique de la mise à jour des ports. Bien entendu il eu été souhaitable de développer encore mais je pense qu'un excès aurait été nuisible à la concision de l'exposé. Vous aurez remarqué que je ne suis pas entré dans le débat de savoir si l'un des systèmes était meilleurs que l'autre. S'il est vrai que je n'utilise actuellement que FreeBSD je n'oublie jamais que ce que je sais du monde UNIX a été appris en étudiant le fonctionnement de LINUX.