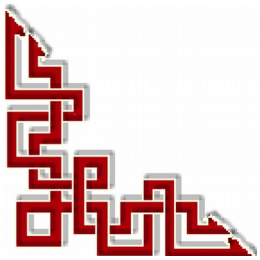
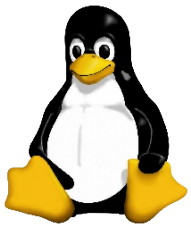
RF-232

Micronator

boot2docker

Les volumes





Version: 0.0.1 / samedi 4 avril 2015 - 16:53

Licence publique générale GNU

AVIS DE NON-RESPONSABILITÉ

Ce document est uniquement destiné à informer. Les informations, ainsi que les contenus et fonctionnalités de ce document sont fournis sans engagement et peuvent être modifiés à tout moment. *RF-232* n'offre aucune garantie quant à l'actualité, la conformité, l'exhaustivité, la qualité et la durabilité des informations, contenus et fonctionnalités de ce document. L'accès et l'utilisation de ce document se font sous la seule responsabilité du lecteur ou de l'utilisateur.

RF-232 ne peut être tenu pour responsable de dommages de quelque nature que ce soit, y compris des dommages directs ou indirects, ainsi que des dommages consécutifs résultant de l'accès ou de l'utilisation de ce document ou de son contenu.

Chaque internaute doit prendre toutes les mesures appropriées (*mettre à jour régulièrement son logiciel antivirus, ne pas ouvrir des documents suspects de source douteuse ou non connue*) de façon à protéger le contenu de son ordinateur de la contamination d'éventuels virus circulant sur la Toile.

Avertissement

Bien que nous utilisions ici un vocabulaire issu des techniques informatiques, nous ne prétendons nullement à la précision technique de tous nos propos dans ce domaine.

Sommaire

I- Description générale

1. Introduction

Ce document préliminaire, le sixième de la série **boot2docker**, décrit la gestion des volumes de **boot2docker**, un système d'exploitation complet qui ne nécessite aucune machine hôte pour tourner et gère les **images** et les **conteneurs** Docker.

Présentement boot2docker, compilé en 32 bits, ne fonctionne qu'avec des systèmes 64 bits.

1.1. Premier document

Pour voir le premier document de la série boot2docker: **http://www.micronator.org/?page_id=1826**.

Pour voir le deuxième document de la série boot2docker: **http://www.micronator.org/?page_id=1875**.

Pour voir le troisième document de la série boot2docker: **http://www.micronator.org/?page_id=1895**.

Pour voir le quatrième document de la série boot2docker: **http://www.micronator.org/?page_id=1913**.

Pour voir le cinquième document de la série boot2docker: **http://www.micronator.org/?page_id=1922**.

2. Docker

Référence: **http://fr.wikipedia.org/wiki/Docker_%28logiciel%29**.

Docker est un **logiciel** **open source** qui automatise le déploiement d'applications dans des conteneurs logiciels. Selon la firme de recherche sur l'industrie, 451 Research, "Docker est un outil qui peut empaqueter une application et ses dépendances dans un conteneur virtuel qui pourra être exécuté sur n'importe quel serveur Linux". Ceci permet d'étendre la flexibilité et la portabilité d'exécution d'une application, que ce soit sur la machine locale, un cloud privé ou public, une machine nue, etc.

Docker étend le format de Conteneur Linux standard, **LXC**, avec une **API** de haut niveau fournissant une solution de virtualisation qui exécute les processus de façon isolée. Docker utilise LXC, **cgroups**, et le **noyau Linux** lui-même. Contrairement aux machines virtuelles traditionnelles, un conteneur Docker n'inclut pas de système d'exploitation, à la place il s'appuie sur les fonctionnalités du système d'exploitation fournies par l'infrastructure sous-jacente.

La technologie de conteneur de Docker peut être utilisée pour étendre des systèmes distribués de façon à ce qu'ils s'exécutent de manière autonome depuis une seule machine physique ou une seule instance par nœud; ce qui permet aux nœuds d'être déployés au fur et à mesure que les ressources sont disponibles, offrant un déploiement transparent et similaire aux **Paas** pour des systèmes comme **Apache Cassandra**, **Riak** ou d'autres systèmes distribués.

2.1. Histoire

Docker a été développé comme un projet interne de dotCloud par Solomon Hykes, une société proposant une **Plate-forme en tant que service**, avec les contributions d'Andrea Luzzardi et Francois-Xavier Bourlet, également employés de dotCloud. Docker est une évolution basée sur les technologies propriétaires de dotCloud, elles-mêmes construites sur des projets open-sources tels que Cloudlets.

Docker a été distribué en tant que projet open source à partir de mars 2013.

Au 18 novembre 2013, le projet a été mis en favoris plus de 7 300 fois sur **GitHub** (14e projet le plus populaire), avec plus de 900 forks et 200 contributeurs.

Au 9 mai 2014, le projet a été mis en favoris plus de 11 769 fois sur **GitHub**, avec plus de 1 912 forks et 423 contributeurs.

3. Logiciels recommandés

3.1. VirtualBox

Logiciel de virtualisation: **https://www.virtualbox.org/**.

3.2. SME-9/64

Système d'exploitation Linux: **http://wiki.contribs.org/SME_Server:Download**.

3.3. digestIT 2004

Logiciel de calcul de somme de contrôle: **http://www.colonywest.us/digestit/**.

3.4. Notepad++

Éditeur de texte ASCII: **http://notepad-plus-plus.org/fr/**.

3.5. WinSCP

Logiciel de téléversement: **http://winscp.net/eng/docs/lang:fr**.

3.6. PuTTY

Logiciel d'accès **SSH**: **http://www.putty.org/**

4. Particularités de ce document

4.1. Notes au lecteur

* Les captures d'écrans ne sont que des références.

** Les informations écrites ont préséance sur celles retrouvées dans les captures d'écrans. Veiller à se référer aux différents tableaux lorsque ceux-ci sont présents.

4.2. Conventions

Toutes les commandes à entrer à la console sont en **gras**. Les affichages à surveiller sont en **rouge**, **bleu**, **orange** ou **magenta**.

```
# ping 192.168.1.149
192.168.1.149 is alive
#
```

Les liens de référence internet sont en **bleu** et ceux intra document en ***bleu***.


Manipulation, truc ou ruse pour se tirer d'embarras.


Une recommandation ou astuce.


Une note.


Une étape, note ou procédure à surveiller.


Paragraphe non complété ou non vérifié.


Cet icône indique que cette commande est sur une seule ligne. Le **PDF** la mettra sur deux lignes avec un [**CR**] [**LF**] entre les deux. Il faudra donc copier la commande entière dans un éditeur de texte ASCII et la mettre sur une seule ligne avant de la copier à la console.

5. Commentaires et suggestions

RF-232 apprécie énormément échanger avec ses internautes. Vos commentaires et suggestions sont indispensables à l'amélioration de la documentation et du site **micronator.org**.

N'hésitez pas à nous transmettre vos commentaires et à nous signaler tout problème d'ordre technique que vous avez rencontré ou n'arrivez pas à résoudre. Tous vos commentaires seront pris en considération et nous vous promettons une réponse dans les plus brefs délais.

II- Gestion des données dans les conteneurs

1. Introduction

Jusqu'ici, nous avons été introduits à certains concepts de base de Docker: les images Docker, l'adressage réseau et les liaisons inter-conteneurs. Dans ce chapitre, nous allons discuter de la gestion des données à l'intérieur et entre les conteneurs Docker.

Nous allons examiner les deux principales gestions de données de Docker.

- La gestion des volumes de données.
- La gestion des conteneurs-volume-de-données.

2. Volumes de données

Un volume de données est un répertoire spécialement assigné à un ou plusieurs conteneurs et qui contourne Union File System. Les volumes de données offrent plusieurs fonctionnalités pour les données persistantes ou partagées.

- Les volumes sont initialisés lorsqu'un conteneur est créé. Si l'image de base du conteneur contient des données au point de montage spécifié, les données sont copiées dans le nouveau volume.
- Les volumes de données peuvent être partagés et réutilisés entre conteneurs.
- Les modifications apportées à un volume de données sont directement effectuées.
- Ces modifications ne seront pas incluses lors d'une mise à jour d'une image.
- Les volumes de données persistent même si le conteneur lui-même est supprimé.

Les volumes de données sont conçus pour garder en permanence les données, indépendamment du cycle de vie du conteneur. Ainsi, Docker ne supprime jamais automatiquement les volumes lorsqu'on supprime un conteneur ni n'exécute de **ramasse-miettes** sur les volumes qui ne sont plus référencés par un conteneur.

3. Création d'un volume de données

Vous pouvez ajouter un volume de données à un conteneur en utilisant l'argument **-v** dans les commandes **docker create** et **docker run**.

Vous pouvez utiliser plusieurs fois l'argument **-v** pour monter plusieurs volumes de données.

Montons un seul volume dans notre conteneur d'application Web.

```
root@boot2docker:~# docker run -d -P --name web -v /webapp  training/webapp python app.py

FATA[0000] Error response from daemon: Conflict. The name "web" is already in use by
container 9a5ffe12a0d9. You have to delete (or rename) that container to be able to reuse
that name.
root@boot2docker:~#
```

Le nom **web** est déjà utilisé par le conteneur **9a5ffe12a0d9**. Il nous faut donc arrêter et détruire ce conteneur avant d'en créer un nouveau utilisant le même nom.

```
root@boot2docker:~# docker stop web

web
root@boot2docker:~#
```

```
root@boot2docker:~# docker rm web

web
root@boot2docker:~#
```

On essaie encore la création du conteneur.

```
root@boot2docker:~# docker run -d -P --name web -v /webapp  training/webapp python app.py

fdfc749fc7060a21a67e6748e113575b34033f5fa7a5815a388d4e04b7dac158
root@boot2docker:~#
```

La commande a créé un nouveau volume: **/webapp** à l'intérieur du conteneur **web**.

Vous pouvez également utiliser l'instruction de volume dans un **fichier Dockerfil**e afin d'ajouter un ou plusieurs nouveaux volumes à n'importe quel conteneur créé à partir de cette image.

4. Monter un répertoire hôte en tant que volume de données

En plus de créer un volume en utilisant l'option **-v** on peut aussi monter un répertoire de l'hôte dans un conteneur.

Si on utilise boot2docker (*notre cas*), le daemon Docker n'a qu'un accès limité au système de fichiers OSX/Windows (*ce qui n'est pas notre cas vu que tous nos systèmes sont des Linux*). Docker essaie d'auto-partager les répertoires **/Users** (*OSX*) ou **C:\Users** (*Windows*). On peut monter des fichiers de ces répertoires ou ces répertoires eux-même à l'aide de **docker run -v /Users/<path>:/<container path> ...** (*OSX*) ou **docker run -v /c/Users/<path>:/<container path ...** (*Windows*). Tous les autres **chemins** proviennent du système de fichiers de la machine virtuelle qui roule Docker.

Pour la même raison que précédemment, si on veut utiliser le nom **web** pour un nouveau conteneur, il nous faut arrêter et supprimer le conteneur qui utilise présentement ce nom.

On arrête le conteneur nommé **web** lancé auparavant.

```
root@boot2docker:~# docker stop web

web
root@boot2docker:~#
```

Vu que le dernier conteneur **web** référençait un volume, il nous faut utiliser un argument supplémentaire "**-v**" pour le détruire. Pour plus de détails, voir l'explication ***Mise en garde importante*** à la page ***12***.

```
root@boot2docker:~# docker rm -v web

web
root@boot2docker:~#
```

On peut maintenant lancer la commande ci-dessous qui utilise encore **web** pour nommer le nouveau conteneur.

```
root@boot2docker:~# docker run -d                              \
                                -P                             \
                                --name web                     \
                                -v /src/webapp:/opt/webapp     \
                                training/webapp python app.py

8bfa677dd9bd6632a0700f5d65e45bf9155a36c1007644518a76ff677568e49d
root@boot2docker:~#
```

Cette commande montera le répertoire source **/src/webapp** du serveur hôte boot2docker dans le répertoire de destination **/opt/webapp** du conteneur **web**.

- Si le point de montage **/opt/webapp** existe déjà à l'intérieur de l'image **training/webapp** du conteneur **web**, son contenu sera remplacé par le contenu du répertoire source **/src/webapp** de l'hôte boot2docker afin de rester cohérent avec le comportement conventionnel de **mount**.
- Le **répertoire** de l'hôte doit être spécifié avec un chemin absolu.
- Si le répertoire n'existe pas sur l'hôte, Docker le créera automatiquement pour nous.
- Ce genre de montage d'un volume est très utile pour le développement logiciel car on peut monter notre code source à l'intérieur du conteneur de destination et voir notre application au travail lorsqu'on change le contenu de la source.

Cette fonctionnalité de montage de volume n'est pas disponible à partir d'un **fichier Dockerfile** en raison de la portabilité et de la finalité du partage d'images construites. Le répertoire de l'hôte est, de par sa nature, dépendant de l'hôte; un répertoire d'hôte spécifié dans un fichier Dockerfile ne fonctionnerait probablement pas sur tous les hôtes.

Un volume Docker est par défaut en **mode lecture-écriture**. Par contre, on peut monter un volume en mode lecture seule, "**ro**" (*read only*).

Encore une fois, on arrête et on supprime le conteneur **web** lancé précédemment.

On lance la commande suivante.

```
root@boot2docker:~# docker run -d                                \
                                -P                               \
                                 --name web                      \
                                 -v /src/webapp:/opt/webapp:ro   \
                                 training/webapp                 \
                                 python app.py

646f59bca91ccfcd6e6f7ca75457742bc7ec09940f4d006f67d66bccf28a30c1
root@boot2docker:~#
```

La commande a monté le même répertoire **/src/webapp** mais on a ajouté l'argument **ro** pour préciser que le **mount** doit être en lecture seule.

On arrête et on supprime le conteneur **web** qu'on vient de lancer.

5. Monter un fichier de l'hôte en tant que volume de données

L'argument **-v** peut également être utilisé pour monter un fichier au lieu d'un répertoire du serveur hôte.

Le fichier de l'historique du shell **ash** sur le serveur boot2docker est: **/root/.ash_history**.

Le fichier de l'historique du shell **bash** sur le serveur **ubuntu:latest** est: **/root/.bash_history**.

On lance la commande.

```
root@boot2docker:~# docker run -i                                          \
                               -t                                          \
                               -v /root/.ash_history:/root/.bash_history   \
                               ubuntu:latest                               \
                               /bin/bash

root@0b9333aa9e75:/#
```

Cette commande va:

- **-i**, lancer le nouveau conteneur en mode **i**nteractive
- **-t,** offrir un **t**erminal
- **-v**, mapper le fichier source **/root/.ash_history** du serveur boot2docker sur le fichier de destination **/root/.bash_history** du conteneur
- **ubuntu:latest**, créer un conteneur avec l'image la plus récente d'Ubuntu
- **/bin/bash**, lancer un **shell bash** à l'intérieur du nouveau conteneur
- ajouter au fichier source de l'hôte i.e. **/root/.ash_history** toutes les commandes lancées à l'intérieur du conteneur, en passant par le fichier **/bin/bash** du conteneur i.e. **/root/.bash_history**.

Conséquences:

- L'historique de l'hôte sera disponible à l'intérieur du conteneur.
- Après la sortie, l'historique du conteneur pourra être consultée dans l'historique de l'hôte.

De nombreux outils utilisés pour modifier des fichiers tels **vi** et **sed** peuvent entraîner une modification de l'**inode**. Depuis **Docker v1.1.0**, ces modifications de fichiers vont produire: **erreur:"sed: ne peut pas renommer ./sedKdJ9Dy: Périphérique ou ressource occupé"**. Au cas où vous souhaiteriez modifier le fichier monté, il est souvent préférable de monter le répertoire parent du fichier.

Nous vous laissons l'expérimentation de cette procédure.

Avant de passer au prochain chapitre, on arrête et on supprime tous les conteneurs qui roulent présentement.

III- Conteneur-volume-de-données

1. Introduction

Si vous avez des données persistantes que vous souhaitez partager avec des conteneurs ou si vous voulez utiliser des données de conteneurs non persistants, il est préférable de créer et nommer un conteneur-volume-de-données, puis de monter les données qu'il contient.

2. Mise en garde importante

Pour supprimer les volumes d'un disque, vous devez lancer explicitement **docker rm -v** lors de la suppression du dernier conteneur, y compris le conteneur initial, ayant une référence au volume.

Lors de la suppression du dernier conteneur, l'argument **-v** permet de mettre à jour ou de migrer efficacement des volumes de données entre conteneurs.

Docker ne vous avertira pas lorsque vous supprimez un conteneur sans utiliser l'argument **-v**. Si vous supprimez des conteneurs sans utiliser l'argument **-v**, vous pouvez vous retrouver avec des "**dangling volumes**" qui ne sont plus référencés par un conteneur. Les "**dangling volumes**" sont difficiles à supprimer et peuvent prendre une grande quantité d'espace disque. Les développeurs de Docker travaillent à l'amélioration de la gestion de volumes et vous pouvez vérifier les progrès à ce sujet en vous référant à: **requête #8484**.

3. Création d'un conteneur-volume-de-données

Créons un nouveau conteneur nommé **conteneurdata** ayant un volume **voldata** à partager.

Bien que ce conteneur ne roule aucune application, nous allons ré-utiliser l'image **training/postgres**. De la sorte, tous les conteneurs utiliseront des couches communes qui résultera en une économie de l'espace disque.

```
root@boot2docker:~# docker create -v /voldata --name conteneurdata training/postgres

ee47972d16fe36cfc51da512a8cddd2a85abb076a34496ca35e6a350e17a9409
root@boot2docker:~#
```

Le volume **/voldata** du nouveau conteneur **conteneurdata** est maintenant offert en partage.

On peut maintenant utiliser l'argument **--volumes-from** pour monter le volume offert en partage dans un premier conteneur: **conteneurdata01**.

```
docker@boot2docker:~$ docker run -d                                \
                                 --volumes-from conteneurdata      \
                                 --name conteneurdata01            \
                                 training/postgres

6338265b4710b85fa61c6725e404766fe265d7183e5957ebb549a3b376480800
docker@boot2docker:~$
```

On monte aussi le volume **/voldata** dans un deuxième conteneur: **conteneurdata02**.

```
docker@boot2docker:~$ docker run -d                              \
                                 --volumes-from conteneurdata    \
                                 --name conteneurdata02          \
                                 training/postgres

73e53374c63db4ef75f2c07e7c02dcce652f99fcdd742098f12cf9a21d1379e5
docker@boot2docker:~$
```

Dans ces deux derniers cas, si l'image originale **training/postgres** (*celle utilisée pour créer tous nos conteneur*) contenait déjà un répertoire appelé **/voldata** et qu'on monte **/voldata** depuis **conteneurdata**, **/voldata** dissimulera les fichiers **/voldata** de l'image **training/postgres** originale. Le résultat est que les fichiers de **/voldata** à partir du conteneur **conteneurdata** sont visibles.

Vous pouvez utiliser plusieurs fois l'argument **--volumes-from** pour réunir plusieurs volumes de données à partir de plusieurs conteneurs.

On peut également étendre la chaîne en montant le volume partagé du conteneur **conteneurdata** sur un troisième conteneur **conteneurdata03** via l'intermédiaire de **conteneurdata01** ou **conteneurdata02**.

```
docker@boot2docker:~$ docker run -d                              \
                                 --volumes-from conteneurdata02  \
                                 --name conteneurdata03          \
                                 training/postgres

fe06f22ba049b1491494e3eff90a2947b99d7c5ab7719052f5b9b62cf8280bd9
docker@boot2docker:~$
```

Si vous supprimez des conteneurs qui montent des volumes, y compris le conteneur initial **conteneurdata** ou les conteneurs subséquents **conteneurdata01** et **conteneurdata02**, les volumes ne seront pas supprimés. Se référer au paragraphe ***Mise en garde importante*** ci-dessus.

4. Suppression de conteneurs avec volume

Nous allons arrêter et supprimer tous les conteneurs.

On arrête d'abord les trois conteneurs.

```
root@boot2docker:~# docker stop conteneurdata01 conteneurdata02

conteneurdata01
conteneurdata02
conteneurdata03
root@boot2docker:~#
```

On supprime seulement les deux premiers conteneurs de manière conventionnelle.

```
root@boot2docker:~# docker rm conteneurdata01  conteneurdata02

conteneurdata01
conteneurdata02
root@boot2docker:~#
```

On supprime, sans oublier l'argument **-v**, le dernier conteneur i.e. **conteneurdata03** ayant monté le volume.

```
root@boot2docker:~# docker rm -v  conteneurdata03

conteneurdata03
root@boot2docker:~#
```

On conserve **conteneurdata** pour le prochain chapitre.

IV- Sauvegarder, restaurer et migrer des volumes

1. Rappel de la mise en garde

Pour supprimer les volumes d'un disque, vous devez lancer explicitement **docker rm -v** lors de la suppression du dernier conteneur, y compris le conteneur initial, ayant une référence au volume.

2. Contenu du volume

Nous allons examiner le contenu du volume **voldata** afin de pouvoir le comparer après sa modification et lors de sa restauration.

```
root@boot2docker:~# docker run -i -t                          \
                            --volumes-from conteneurdata      \
                            --name conteneurdata04            \
                            training/postgres                 \
                            /bin/bash

root@7306b28b82db:/#
```

Une fois à l'intérieur du nouveau conteneur, nous affichons le contenu de **/voldata** du conteneur nouvellement créé.

```
root@7306b28b82db:/# ls -als /voldata/

total 8
4 drwxr-xr-x  2 root root 4096 Mar 28 14:26 .
4 drwxr-xr-x 61 root root 4096 Mar 28 14:27 ..
root@7306b28b82db:/#
```

Le répertoire est vide.

2.1. Modification

Nous allons y créer un fichier pour notre comparaison ultérieure.

```
root@7306b28b82db:/# touch /voldata/fichier_original_dans_voldata

root@7306b28b82db:/#
```

On vérifie.

```
root@7306b28b82db:/# ls -als /voldata/

total 8
4 drwxr-xr-x  2 root root 4096 Mar 28 14:28 .
4 drwxr-xr-x 61 root root 4096 Mar 28 14:27 ..
0 -rw-r--r--  1 root root    0 Mar 28 14:28 fichier_original_dans_voldata
root@7306b28b82db:/#
```

On sort du conteneur.

```
root@7306b28b82db:/# exit

exit
root@boot2docker:~#
```

2.1.1. Vérification

On lance un nouveau conteneur pour vérifier si le fichier est toujours là.

```
root@boot2docker:~# docker run -i -t                          \
                              --volumes-from conteneurdata    \
                              --name conteneurdata05          \
                               training/postgres              \
                               /bin/bash

root@boot2docker:~#
```

```
root@boot2docker:~# ls -als /voldata/

total 8
4 drwxr-xr-x  2 root root 4096 Mar 28 14:28 .
4 drwxr-xr-x 61 root root 4096 Mar 28 14:30 ..
0 -rw-r--r--  1 root root    0 Mar 28 14:28 fichier_original_dans_voldata
root@0486d1d8e311:/#
```

Il y est.

On sort du conteneur.

```
root@0486d1d8e311:/# exit

exit
root@boot2docker:~#
```

3. Sauvegarde d'un volume

D'autres fonctions très utiles que nous pouvons effectuer avec les volumes est de les utiliser pour des sauvegardes, restaurations ou migrations.

3.1. Création d'un conteneur qui sauvegarde un volume

Pour ce faire, on crée un nouveau conteneur qui exécute la sauvegarde du volume.

```
root@boot2docker:~# docker run --volumes-from conteneurdata                  \
                               -v $(pwd):/sauvegarde                         \
                               --name conteneur06                            \
                               ubuntu                                        \
                               tar -cvf /sauvegarde/sauvegarde.tar /voldata

tar: Removing leading `/' from member names
/voldata/
/voldata/fichier_original_dans_voldata
root@boot2docker:~# ls -als
```

--volumes-from conteneurdata Le nouveau conteneur montera le volume **/voldata** de **conteneurdata**.

-v $(pwd):/sauvegarde Il montera aussi le répertoire actuel **$(pwd)** de l'hôte dans le répertoire **/sauvegarde** du nouveau conteneur.

--name conteneur06 Le nouveau conteneur se nommera conteneur **conteneur06**.

ubuntu Il utilisera l'image ubuntu.

tar -cvf /sauvegarde/sauvegarde.tar /voldata

À l'intérieur du conteneur, on exécutera un **tar:**

c - **c**réer un fichier **tar**

v - en mode **v**erbose i.e. afficher tout ce que fait le **tar**.

f - le **f**ichier de sortie sera **/sauvegarde/sauvegarde .tar**

faire un **tar** du répertoire **/voldata**.

Lorsque ces instructions seront terminées, le conteneur s'arrêtera vu qu'il n'aura plus d'autres instructions à exécuter.

Après la sortie, à l'invite du serveur hôte, on affiche le contenu du répertoire courant.

```
root@boot2docker:~# ls -ls

total 12
    12 -rw-r--r--    1 root     staff        10240 Mar 28 15:46 sauvegarde.tar
root@boot2docker:~#
```

Nous voyons que le **tar** a bien créé le fichier **sauvegarde.tar** dans le répertoire courant.

3.2. Vérification

Pour vérifier le fichier **sauvegarde.tar**, on crée un répertoire temporaire **/temp** on y extrait le fichier.

```
root@boot2docker:~# mkdir /temp

root@boot2docker:~#
```

```
root@boot2docker:~# tar -xvf sauvegarde.tar -C /temp

voldata/
voldata/fichier_original_dans_voldata
root@boot2docker:~#
```

On vérifie.

```
root@boot2docker:~# ls -als /temp/

total 0
     0 drwxr-xr-x    3 root     root       60 Mar 28 16:55 .
     0 drwxr-xr-x   17 root     root      400 Mar 28 16:52 ..
     0 drwxr-xr-x    2 root     root       60 Mar 28 14:28 voldata
root@boot2docker:~#
```

```
root@boot2docker:~# ls -als /temp/voldata/

total 0
     0 drwxr-xr-x    2 root     root       60 Mar 28 14:28 .
     0 drwxr-xr-x    3 root     root       60 Mar 28 16:55 ..
     0 -rw-r--r--    1 root     root        0 Mar 28 14:28 fichier_original_dans_voldata
root@boot2docker:~#
```

4. Ajout au volume

On modifie encore une fois le contenu du volume en y ajoutant un nouveau fichier pour simuler un travail d'ajout.

```
root@boot2docker:~# docker run -i -t                          \
                           --volumes-from conteneurdata       \
                           --name conteneur07                 \
                           training/postgres                  \
                           touch /voldata/fichier_d-ajout_dans_voldata

root@boot2docker:~#
```

On vérifie le contenu du volume.

```
root@boot2docker:~# docker run -i -t                          \
                            --volumes-from conteneurdata      \
                            --name conteneur08                \
                            training/postgres                 \
                            ls -als /voldata/

total 8
4 drwxr-xr-x  2 root root 4096 Mar 28 17:09 .
4 drwxr-xr-x 61 root root 4096 Mar 28 17:14 ..
0 -rw-r--r--  1 root root    0 Mar 28 17:09 fichier_d-ajout_dans_voldata
0 -rw-r--r--  1 root root    0 Mar 28 14:28 fichier_original_dans_voldata
root@boot2docker:~#
```

Le fichier d'ajout a bien été créé.

5. Restauration d'un volume à son emplacement original

Nous nous apercevons que la dernière modification est une erreur. Nous devons restaurer notre sauvegarde.

Nous allons créer un **script bash** qui roulera à l'intérieur d'un nouveau conteneur qu'on créera.

Ce script:

- effacera le contenu du volume
- affichera le volume qui doit maintenant être vide
- fera la restauration de la sauvegarde
- enfin, il affichera le contenu restauré du volume.

On crée le script.

```
cat > /root/restaure.sh << FIN
#!/bin/bash

# Nous effaçons complètement le contenu du répertoire /voldata
/bin/rm -rf /voldata/*

# On vérifie.
/bin/ls -als /voldata

# Le fichier de sauvegarde est: /sauvegarde/sauvegarde.tar
# On l'extrait dans / car les fichiers dans le tar contiennent leur chemin
/bin/tar -xvf /sauvegarde/sauvegarde.tar -C /

# On affiche le répertoire restauré
/bin/ls -als /voldata/

FIN
```

On rend le script exécutable par le propriétaire du script seulement.

```
root@boot2docker:~# chmod u+x restaure.sh

root@boot2docker:~#
```

On vérifie les droits et privilèges du script.

```
root@boot2docker:~# ls -ls restaure.sh

     4 -rwxr--r--    1 root     root           343 Mar 28 19:25 restaure.sh
root@boot2docker:~#
```

On vérifie le contenu du script.

La première ligne du script n'est pas vide. Ici la ligne vide sert seulement à faciliter la copie de la commande **cat restaure.sh** depuis le fichier **PDF** vers l'invite du serveur. La première ligne d'un script bash doit toujours débuter par #!/bin/bash.

```
root@boot2docker:~# cat restaure.sh

#!/bin/bash

# Nous effaçons complètement le contenu du répertoire /voldata
/bin/rm -rf /voldata/*

# On vérifie.
/bin/ls -als /voldata

# Le fichier de sauvegarde est: /sauvegarde/sauvegarde.tar
# On l'extrait dans / car les fichiers dans le tar contiennent leur chemin
/bin/tar -xvf /sauvegarde/sauvegarde.tar -C /

# On affiche le répertoire restauré
/bin/ls -als /voldata/

root@boot2docker:~#
```

Nous allons monter le répertoire courant de l'hôte dans le répertoire **/sauvegarde** du conteneur. De cette façon. le shell du conteneur aura accès au script **restaure.sh** et au fichier de sauvegarde **sauvegarde.tar**.

Nous allons utiliser les arguments **-i** et **-t** de même que **--rm** qui lui, effacera le conteneur à la sortie.

```
root@boot2docker:~# docker run  -i -t                                   \
                                --rm                                    \
                                --volumes-from conteneurdata            \
                                -v $(pwd):/sauvegarde                   \
                                --name conteneur09                      \
                                ubuntu:latest                           \
                                /bin/bash /sauvegarde/restaure.sh

total 8
4 drwxr-xr-x  2 root root 4096 Mar 28 19:35 .
4 drwxr-xr-x 34 root root 4096 Mar 28 19:35 ..
voldata/
voldata/fichier_original_dans_voldata
voldata/fichier_d-ajout_dans_voldata
total 8
4 drwxr-xr-x  2 root root 4096 Mar 28 19:32 .
4 drwxr-xr-x 34 root root 4096 Mar 28 19:35 ..
0 -rw-r--r--  1 root root    0 Mar 28 19:32 fichier_original_dans_voldata
root@boot2docker:~#
```

Le script **restaure.sh** a été exécuté à l'intérieur du conteneur.

On peut vérifier le contenu du volume **voldata**.

```
root@boot2docker:~# docker run  -i -t                                   \
                                --rm                                    \
                                --volumes-from conteneurdata            \
                                --name conteneur09                      \
                                ubuntu:latest                           \
                                ls -als /voldata

total 8
4 drwxr-xr-x  2 root root 4096 Mar 28 19:32 .
4 drwxr-xr-x 33 root root 4096 Mar 28 19:44 ..
0 -rw-r--r--  1 root root    0 Mar 28 19:32 fichier_original_dans_voldata
root@boot2docker:~#
```

On peut conclure que la restauration a réussi. Le fichier ajouté par erreur est disparu.

6. Restauration d'un volume à un nouvel emplacement

Pour restaurer un sauvegarde dans un nouveau conteneur:

- on crée un nouveau conteneur
- on y monte le répertoire qui contient la sauvegarde en tant que volume.
- une fois à l'intérieur du nouveau conteneur on restaure la sauvegarde en lançant le script de restauration..


Il faut absolument être dans le répertoire contenant la sauvegarde avant de créer le nouveau conteneur.

```
root@boot2docker:~# docker run  -i -t                                   \
                                -v $(pwd):/sauvegarde                   \
                                --name conteneur09                      \
                                ubuntu:latest                           \
                                /bin/bash

root@35b44001840a:/#
```

On vérifie que le répertoire contenant la sauvegarde a bien été monté.

```
root@35b44001840a:/# ls -ls sauvegarde/

total 16
12 -rw-r--r-- 1 root root 10240 Mar 28 20:01 sauvegarde.tar
 4 -rwxr--r-- 1 root root   343 Mar 28 20:05 restaure.sh
root@35b44001840a:/#
```

On restaure la sauvegarde en spécifiant à l'aide de l'argument **-C** du **tar**, le répertoire parent du répertoire de destination de la restauration. Ici, on restaure dans le répertoire racine du conteneur.

```
root@35b44001840a:/# tar -xvf /sauvegarde/sauvegarde.tar -C /

voldata/
voldata/fichier_original_dans_voldata
root@35b44001840a:/#
```

On vérifie le répertoire de restauration.

```
root@35b44001840a:/# ls -alsd /voldata

4 drwxr-xr-x 2 root root 4096 Mar 28 20:00 /voldata
root@35b44001840a:/#
```

On vérifie son contenu.

```
root@35b44001840a:/# ls -ls /voldata

total 0
0 -rw-r--r-- 1 root root 0 Mar 28 19:32 fichier_original_dans_voldata
root@35b44001840a:/#
```

La restauration a fonctionné.

On sort du conteneur.

```
root@35b44001840a:/# exit

exit
root@boot2docker:~#
```

V- Nettoyage du serveur

1. Rappel de la mise en garde

Pour supprimer les volumes d'un disque, vous devez lancer explicitement **docker rm -v** lors de la suppression du dernier conteneur, y compris le conteneur initial, ayant une référence au volume.

Lors de la suppression du dernier conteneur, l'argument **-v** permet de mettre à jour ou de migrer efficacement des volumes de données entre conteneurs.

Docker ne vous avertira pas lorsque vous supprimer un conteneur sans utiliser l'argument **-v**. Si vous supprimez des conteneurs sans utiliser l'argument **-v**, vous pouvez vous retrouver avec des "**dangling volumes**" qui ne sont plus référencés par un conteneur. Les "**dangling volumes**" sont difficiles à supprimer et peuvent prendre une grande quantité d'espace disque. Les développeurs de Docker travaillent à l'amélioration de la gestion de volumes et vous pouvez vérifier les progrès à ce sujet en vous référant à: **requête #8484**.

2. Nettoyage

On vérifie qu'aucun conteneur ne roule, sinon on les arrête avec la commande **docker stop**.

```
root@boot2docker:~# docker ps

CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES
root@boot2docker:~#
```

Tous nos conteneurs se nommaient **conteneurdata0X** ou **conteneur0X**.

On affiche ces conteneurs.

```
root@boot2docker:~# docker ps -a | grep conteneur*

35b44001840a        ubuntu:14.04                "/bin/bash"           22 minutes ago
Exited (130) 7 minutes ago                      conteneur09
add770190172        training/postgres:latest    "ls -als /voldata/"   39 minutes ago
Exited (0) 39 minutes ago                       conteneur08
3690e93cf0bc        training/postgres:latest    "touch /voldata/fich  40 minutes ago
Exited (0) 40 minutes ago                       conteneur07
3f6f050023a0        ubuntu:14.04                "tar -cvf /backup/ba  42 minutes ago
Exited (0) 42 minutes ago                       conteneur06
efd5c0609f28        training/postgres:latest    "/bin/bash"           42 minutes ago
Exited (0) 42 minutes ago                       conteneurdata05
a7ee14970408        training/postgres:latest    "/bin/bash"           44 minutes ago
Exited (0) 43 minutes ago                       conteneurdata04
ae828d46e64c        training/postgres:latest    "su postgres -c '/us  45 minutes ago
conteneurdata
root@boot2docker:~#
```

Tous nos conteneurs qui montait un volume, utilisaient **voldata** de **conteneurdata**. Nous allons donc supprimer tous ces conteurs sauf **conteneur09** et **conteneurdata** qu'on supprimera tous deux en dernier avec l'argument **-v** afin d'éviter les "**dangling volumes**.

```
root@boot2docker:~# docker rm conteneur08 conteneur07 conteneur06 conteneurdata05
conteneurdata04

conteneur08
conteneur07
conteneur06
conteneurdata05
conteneurdata04
root@boot2docker:~#
```

On supprime **conteneur09**, le dernier conteneur référençant **voldata**.

```
root@boot2docker:~# docker rm -v conteneur09

conteneur09
root@boot2docker:~#
```

On supprime **conteneurdata** le conteneur source offrant **voldata** en partage.

```
root@boot2docker:~# docker rm -v conteneurdata

conteneurdata
root@boot2docker:~#
```

On vérifie.

```
root@boot2docker:~# docker ps -a | grep conteneur*

root@boot2docker:~#
```

3. Prochaine étape

Dans ce chapitre, nous avons approfondi la façon d'utiliser Docker. Dans le suivant, nous allons voir comment combiner Docker avec les services disponibles sur **Docker Hub** y compris les **constructions automatiques** et les **registres privés**.

Victoire totale, hissons la bannière de la victoire.

Crédits

Auteur: **Michel-André Robillard CLP**

Remerciement: **Tous les contributeurs GNU/GPL**.

Intégré par: **Michel-André Robillard CLP**

Contact: **michelandre at micronator.org**

Répertoire de ce document: E:\000_DocPourRF232_general\RF-232_Docker\6_Volumes\RF-232_boot2docker_Les-Volumes_2015-04-04_16h51.odt

Historique des modifications**:**

Version	*Date*	*Commentaire*	*Auteur*
RC-1	2015-03-25	Début.	M.-A. Robillard
0.0.1	2015-04-04	Vérification complète et corrections.	M.-A. Robillard

Index

LICENCE PUBLIQUE GÉNÉRALE GNU

Version 3, du 29 juin 2007.

- -

Traduction française par Philippe Verdy <verdy_p (à) wanadoo (point) fr>, le 30 juin 2007 (dernière correction du 4 janvier 2011).

Avertissement important au sujet de cette traduction française.

Ceci est une traduction en français de la licence “GNU General Public License” (GPL). Cette traduction est fournie ici dans l’espoir qu’elle facilitera sa compréhension, mais elle ne constitue pas une traduction officielle ou approuvée d’un point de vue juridique.

La Free Software Foundation (FSF) ne publie pas cette traduction et ne l’a pas approuvée en tant que substitut valide au plan légal pour la licence authentique “GNU General Public Licence”. Cette traduction n’a pas encore été passée en revue attentivement par un juriste et donc le traducteur ne peut garantir avec certitude qu’elle représente avec exactitude la signification légale des termes de la licence authentique “GNU General Public License” publiée en anglais. Cette traduction n’établit donc légalement aucun des termes et conditions d’utilisation d’un logiciel sous licence GNU GPL — seul le texte original en anglais le fait. Si vous souhaitez être sûr que les activités que vous projetez seront autorisées par la GNU General Public License, veuillez vous référer à sa seule version anglaise authentique.

La FSF vous recommande fermement de ne pas utiliser cette traduction en tant que termes officiels pour vos propres programmes; veuillez plutôt utiliser la version anglaise authentique telle que publiée par la FSF. Si vous choisissez d’acheminer cette traduction en même temps qu’un Programme sous licence GNU GPL, cela ne vous dispense pas de l’obligation d’acheminer en même temps une copie de la licence authentique en anglais, et de conserver dans la traduction cet avertissement important en français et son équivalent en anglais ci-dessous.

Important Warning About This French Translation.

This is a translation of the GNU General Public License (GPL) into French. This translation is distributed in the hope that it will facilitate understanding, but it is not an official or legally approved translation.

The Free Software Foundation (FSF) is not the publisher of this translation and has not approved it as a legal substitute for the authentic GNU General Public License. The translation has not been reviewed carefully by lawyers, and therefore the translator cannot be sure that it exactly represents the legal meaning of the authentic GNU General Public License published in English. This translation does not legally state the terms and conditions of use of any Program licenced under GNU GPL — only the original English text of the GNU LGPL does that. If you wish to be sure whether your planned activities are permitted by the GNU General Public License, please refer to its sole authentic English version.

The FSF strongly urges you not to use this translation as the official distribution terms for your programs; instead, please use the authentic English version published by the FSF. If you choose to convey this translation along with a Program covered by the GPL Licence, this does not remove your obligation to convey at the same time a copy of the authentic GNU GPL License in English, and you must keep in this translation this important warning in English and its equivalent in French above.

- -

Préambule

La Licence Publique Générale GNU (**“GNU General Public License”**) est une licence libre, en **“copyleft”**, destinée aux œuvres logicielles et d’autres types d’œuvres.

Les licences de la plupart des œuvres logicielles et autres œuvres de la pratique sont conçues pour vous ôter votre liberté de partager et modifier ces œuvres. À l’inverse, la Licence Publique Générale GNU a pour but de garantir votre liberté de partager et changer toutes les versions d’un programme — afin d’assurer qu’il restera libre pour tous les utilisateurs. Nous, la **Free Software Foundation**, utilisons la Licence Publique Générale GNU pour la plupart de nos logiciels; cela s’applique aussi à toute autre œuvre éditée de cette façon par ses auteurs. Vous pouvez, vous aussi, l’appliquer à vos propres programmes.

Quand nous parlons de logiciel libre (**“free”**), nous nous référons à la liberté (**“freedom”**), pas au prix. Nos Licences Publiques Générales sont conçues pour assurer que vous ayez la liberté de distribuer des copies de logiciel libre (et le facturer si vous le souhaitez), que vous receviez le code source ou pouviez l’obtenir si vous le voulez, que vous pouviez modifier le logiciel ou en utiliser toute partie dans de nouveaux logiciels libres, et que vous sachiez que vous avez le droit de faire tout ceci.

Pour protéger vos droits, nous avons besoin d’empêcher que d’autres vous restreignent ces droits ou vous demande de leur abandonner ces droits. En conséquence, vous avez certaines responsabilités si vous distribuez des copies d’un tel programme ou si vous le modifiez : les responsabilités de respecter la liberté des autres.

Par exemple, si vous distribuez des copies d’un tel programme, que ce soit gratuit ou contre un paiement, vous devez accorder aux Destinataires les mêmes libertés que vous avez reçues. Vous devez aussi vous assurer qu’eux aussi reçoivent ou peuvent recevoir son code source. Et vous devez leur montrer les termes de cette licence afin qu’ils connaissent leurs droits.

Les développeurs qui utilisent la GPL GNU protègent vos droits en deux étapes : (1) ils affirment leur droits d’auteur (**“**copyright**”**) sur le logiciel, et (2) vous accordent cette Licence qui vous donne la permission légale de le copier, le distribuer et/ou le modifier.

Pour la protection des développeurs et auteurs, la GPL stipule clairement qu’il n’y a pas de garantie pour ce logiciel libre. Aux fins à la fois des utilisateurs et auteurs, la GPL requière que les versions modifiées soient marquées comme changées, afin que leurs problèmes ne soient pas attribués de façon erronée aux auteurs des versions précédentes.

Certains dispositifs sont conçus pour empêcher l’accès des utilisateurs à l’installation ou l’exécution de versions modifiées du logiciel à l’intérieur de ces dispositifs, alors que les fabricants le peuvent. Ceci est fondamentalement incompatible avec le but de protéger la liberté des utilisateurs de modifier le logiciel. L’aspect systématique de tels abus se produit dans le secteur des produits destinés aux utilisateurs individuels, ce qui est précisément ce qui est le plus inacceptable. Aussi, nous avons conçu cette version de la GPL pour prohiber cette pratique pour ces produits. Si de tels problèmes surviennent dans d’autres domaines, nous nous tenons prêt à étendre cette restriction à ces domaines dans de futures versions de la GPL, autant qu’il sera nécessaire pour protéger la liberté des utilisateurs.

Finalement, chaque programme est constamment menacé par les brevets logiciels. Les États ne devraient pas autoriser de tels brevets à restreindre le développement et l’utilisation de logiciels libres sur des ordinateurs d’usage général; mais dans ceux qui le font, nous voulons spécialement éviter le danger que les brevets appliqués à un programme libre puisse le rendre effectivement propriétaire. Pour empêcher ceci, la GPL assure que les brevets ne peuvent être utilisés pour rendre le programme non-libre.

Les termes précis et conditions concernant la copie, la distribution et la modification suivent.

TERMES ET CONDITIONS

Article 0. Définitions.

"Cette Licence" se réfère à la version 3 de la **“**GNU General Public License**”** (le texte original en anglais).

"Droit d’Auteur" signifie aussi les droits du “copyright” ou voisins qui s’appliquent à d’autres types d’œuvres, tels que celles sur les masques de semi-conducteurs.

"Le Programme" se réfère à toute œuvre qui peut être sujette au Droit d’Auteur (“copyright”) et dont les droits d’utilisation sont concédés en vertu de cette Licence. Chacun des Licenciés, à qui cette Licence est concédée, est désigné par "vous." Les "Licenciés" et les "Destinataires" peuvent être des personnes physiques ou morales (individus ou organisations).

"Modifier" une œuvre signifie en obtenir une copie et adapter tout ou partie de l’œuvre d’une façon qui nécessite une autorisation d’un titulaire de Droit d’Auteur, autre que celle permettant d’en produire une copie conforme. L’œuvre résultante est appelée une "version modifiée" de la précédente œuvre, ou une œuvre "basée sur" la précédente œuvre.

Une "Œuvre Couverte" signifie soit le Programme non modifié soit une œuvre basée sur le Programme.

"Propager" une œuvre signifie faire quoi que ce soit avec elle qui, sans permission, vous rendrait directement ou indirectement responsable d’un délit de contrefaçon suivant les lois relatives au Droit d’Auteur, à l’exception de son exécution sur un ordinateur ou de la modification d’une copie privée. La propagation inclue la copie, la distribution (avec ou sans modification), la mise à disposition envers le public, et aussi d'autres activités dans certains pays.

"Acheminer" une œuvre signifie tout moyen de propagation de celle-ci qui permet à d’autres parties d’en réaliser ou recevoir des copies. La simple interaction d’un utilisateur à travers un réseau informatique, sans transfert effectif d’une copie, ne constitue pas un acheminement.

Une interface utilisateur interactive affiche des "Notices Légales Appropriées" quand elle comprend un dispositif convenable, bien visible et évident qui (1) affiche une notice appropriée sur les droits d’auteur et (2) informe l’utilisateur qu’il n’y a pas de garantie pour l’œuvre (sauf si des garanties ont été fournies hors du cadre de cette Licence), que les licenciés peuvent acheminer l’œuvre sous cette Licence, et comment consulter une copie de cette Licence. Si l’interface présente une liste de commandes utilisateur ou d’options, tel qu’un menu, un élément évident dans la liste présentée remplit ce critère.

Article 1. Code source.

Le "code source" d’une œuvre signifie la forme préférée de l’œuvre qui permet ou facilite les modifications de celle-ci. Le "code objet" d’une œuvre signifie toute forme de l’œuvre qui n’en est pas le code source.

Une "Interface Standard" signifie une interface qui est soit celle d’une norme officielle définie par un organisme de normalisation reconnu ou, dans le cas des interfaces spécifiées pour un langage de programmation particulier, une interface largement utilisée parmi les développeurs qui travaillent dans ce langage.

Les "Bibliothèques Système" d’une œuvre exécutable incluent tout ce qui, en dehors de l’œuvre dans son ensemble, (a) est inclus dans la forme usuelle de paquetage d’un Composant Majeur mais ne fait pas partie de ce Composant Majeur et (b) sert seulement à permettre l’utilisation de l’œuvre avec ce Composant Majeur ou à mettre en œuvre une Interface Standard pour laquelle une mise en œuvre est disponible au public sous forme de code source; un "Composant Majeur" signifie, dans ce contexte, un composant majeur essentiel (noyau, système de fenêtrage, etc.) du système d’exploitation (le cas échéant) d’un système sur lequel l’œuvre exécutable fonctionne, ou bien un compilateur utilisé pour produire le code objet de l’œuvre, ou un interprète de code objet utilisé pour exécuter celui-ci.

Le "Source Correspondant" d’une œuvre sous forme de code objet signifie l’ensemble des codes sources nécessaires pour générer, installer et (dans le cas d’une œuvre exécutable) exécuter le code objet et modifier l’œuvre, y compris les scripts pour contrôler ces activités. Cependant, cela n’inclue pas les Bibliothèques Système de l’œuvre, ni les outils d’usage général ou les programmes libres généralement disponibles qui peuvent être utilisés sans modification pour achever ces activités mais ne sont pas partie de cette œuvre. Par exemple le Source Correspondant inclut les fichiers de définition d’interfaces associés aux fichiers sources de l’œuvre, et le code source des bibliothèques partagées et des sous-routines liées dynamiquement, pour lesquelles l’œuvre est spécifiquement conçue pour les requérir via, par exemple, des communications de données ou contrôles de flux internes entre ces sous-programmes et d’autres parties de l’œuvre.

Le Source Correspondant n’a pas besoin d’inclure tout ce que les utilisateurs peuvent régénérer automatiquement à partir d’autres parties du Source Correspondant.

Le Source Correspondant pour une œuvre sous forme de code source est cette même œuvre.

Article 2. Permissions de base.

Tous les droits accordés suivant cette Licence le sont jusqu’au terme des Droits d’Auteur (“copyright”) sur le Programme, et sont irrévocables pourvu que les conditions établies soient remplies. Cette Licence affirme explicitement votre permission illimitée d’exécuter le Programme non modifié. La sortie produite par l’exécution d’une Œuvre Couverte n’est couverte par cette Licence que si cette sortie, étant donné leur contenu, constitue une Œuvre Couverte. Cette Licence reconnaît vos propres droits d’usage raisonnable (“fair use” en législation des États-Unis d’Amérique) ou autres équivalents, tels qu’ils sont pourvus par la loi applicable sur le Droit d’Auteur (“copyright”).

Vous pouvez créer, exécuter et propager sans condition des Œuvres Couvertes que vous n’acheminez pas, aussi longtemps que votre licence demeure en vigueur. Vous pouvez acheminer des Œuvres Couvertes à d’autres personnes dans le seul but de leur faire réaliser des modifications à votre usage exclusif, ou pour qu’ils vous fournissent des facilités vous permettant d’exécuter ces œuvres, pourvu que vous vous conformiez aux termes de cette Licence lors de l’acheminement de tout matériel dont vous ne contrôlez pas le Droit d’Auteur (“copyright”). Ceux qui, dès lors, réalisent ou exécutent pour vous les Œuvres Couvertes ne doivent alors le faire qu’exclusivement pour votre propre compte, sous votre direction et votre contrôle, suivant des termes qui leur interdisent de réaliser, en dehors de leurs relations avec vous, toute copie de votre matériel soumis au Droit d’Auteur.

L’acheminement dans toutes les autres circonstances n’est permis que selon les conditions établies ci-dessous. La concession de sous-licences n’est pas autorisé; l’article 10 rend cet usage non nécessaire.

Article 3. Protection des droits légaux des utilisateurs envers les lois anti-contournement.

Aucune Œuvre Couverte ne doit être vue comme faisant partie d’une mesure technologique effective selon toute loi applicable remplissant les obligations prévues à l’article 11 du traité international sur le droit d’auteur adopté à l’OMPI le 20 décembre 1996, ou toutes lois similaires qui prohibent ou restreignent le contournement de telles mesures.

Si vous acheminez une Œuvre Couverte, vous renoncez à tout pouvoir légal d’interdire le contournement des mesures technologiques dans tous les cas où un tel contournement serait effectué en exerçant les droits prévus dans cette Licence pour cette Œuvre Couverte, et vous déclarez rejeter toute intention de limiter l’opération ou la modification de l’Œuvre, en tant que moyens pour renforcer, à l’encontre des utilisateurs de cette Œuvre, vos droits légaux ou ceux de tierces parties d’interdire le contournement desdites mesures technologiques.

Article 4. Acheminement des copies conformes.

Vous pouvez acheminer des copies conformes du code source du Programme tel que vous l’avez reçu, sur n’importe quel support, pourvu que vous publiiez scrupuleusement et de façon appropriée sur chaque copie une notice de Droit d’Auteur appropriée; gardez intactes toutes les notices établissant que cette Licence et tous les termes additionnels non permissifs ajoutés en accord avec l’article 7 s’appliquent à ce code; et donnez à chacun des Destinataires une copie de cette Licence en même temps que le Programme.

Vous pouvez facturer un prix quelconque, y compris gratuit, pour chacune des copies que vous acheminez, et vous pouvez offrir une protection additionnelle de support ou de garantie en échange d’un paiement.

Article 5. Acheminement des versions sources modifiées.

Vous pouvez acheminer une œuvre basée sur le Programme, ou bien les modifications pour le produire à partir du Programme, sous la forme de code source suivant les termes de l’article 4, pourvu que vous satisfassiez aussi à chacune des conditions requises suivantes :

- a) L’œuvre doit comporter des notices évidentes établissant que vous l’avez modifiée et donnant la date correspondante.
- b) L’œuvre doit comporter des notices évidentes établissant qu’elle est éditée selon cette Licence et les conditions ajoutées d’après l’article 7. Cette obligation vient modifier l’obligation de l’article 4 de "garder intactes toutes les notices."
- c) Vous devez licencier l’œuvre entière, comme un tout, suivant cette Licence à quiconque entre en possession d’une copie. Cette Licence s’appliquera en conséquence, avec les termes additionnels applicables prévus par l’article 7, à la totalité de l’œuvre et chacune de ses parties, indépendamment de la façon dont elles sont empaquetées. Cette licence ne donne aucune permission de licencier l’œuvre d’une autre façon, mais elle n’invalide pas une telle permission que vous auriez reçue séparément.
- d) Si l’œuvre a des interfaces utilisateurs interactives, chacune doit afficher les Notices Légales Appropriées; cependant si le Programme a des interfaces qui n’affichent pas les Notices Légales Appropriées, votre œuvre n’a pas à les modifier pour qu’elles les affichent.

Une compilation d’une Œuvre Couverte avec d’autres œuvres séparées et indépendantes, qui ne sont pas par leur nature des extensions de l’Œuvre Couverte, et qui ne sont pas combinés avec elle de façon à former un programme plus large, dans ou sur un volume de stockage ou un support de distribution, est appelé un "agrégat" si la compilation et son Droit d’Auteur résultant ne sont pas utilisés pour limiter l’accès ou les droits légaux des utilisateurs de la compilation en deçà de ce que permettent les œuvres individuelles. L’inclusion d’une Œuvre Couverte dans un agrégat ne cause pas l’application de cette Licence aux autres parties de l’agrégat.

Article 6. Acheminement des formes non sources.

Vous pouvez acheminer sous forme de code objet une Œuvre Couverte suivant les termes des articles 4 et 5, pourvu que vous acheminiez également suivant les termes de cette Licence le Source Correspondant lisible par une machine, d’une des façons suivantes :

- a) Acheminer le code objet sur, ou inclus dans, un produit physique (y compris un support de distribution physique), accompagné par le Source Correspondant fixé sur un support physique durable habituellement utilisé pour les échanges de logiciels.
- b) Acheminer le code objet sur, ou inclus dans, un produit physique (y compris un support de distribution physique), accompagné d’une offre écrite, valide pour au moins trois années et valide pour aussi longtemps que vous fournissez des pièces de rechange ou un support client pour ce modèle de produit, afin de donner à quiconque possède le code objet soit (1) une copie du Source Correspondant à tout logiciel dans ce produit qui est couvert par cette Licence, sur un support physique durable habituellement utilisé pour les échanges de logiciels, pour un prix non supérieur au coût raisonnable de la réalisation physique de l’acheminement de la source, ou soit (2) un accès permettant de copier le Source Correspondant depuis un serveur réseau sans frais.
- c) Acheminer des copies individuelles du code objet avec une copie de l’offre écrite de fournir le Source Correspondant. Cette alternative est permise seulement occasionnellement et non-commercialement, et seulement si vous avez reçu le code objet avec une telle offre, en accord avec l’article 6 alinéa b.
- d) Acheminer le code objet en offrant un accès depuis un emplacement désigné (gratuit ou contre facturation) et offrir un accès équivalent au Source Correspondant de la même façon via le même emplacement et sans facturation supplémentaire. Vous n’avez pas besoin d’obliger les Destinataires à copier le Source Correspondant en même temps que le code objet. Si l’emplacement pour copier le code objet est un serveur réseau, le Source Correspondant peut être sur un serveur différent (opéré par vous ou par un tiers) qui supporte des facilités équivalentes de copie, pourvu que vous mainteniez des directions claires à proximité du code objet indiquant où trouver le Source Correspondant. Indépendamment de quel serveur héberge le Source Correspondant, vous res-

tez obligé de vous assurer qu'il reste disponible aussi longtemps que nécessaire pour satisfaire à ces obligations.

- e) Acheminer le code objet en utilisant une transmission d'égal-à-égal, pourvu que vous informiez les autres participants sur l'endroit où le code objet et le Source Correspondant de l'œuvre sont offerts sans frais au public général suivant l'article 6 alinéa d.

Une portion séparable du code objet, dont le code source est exclu du Source Correspondant en tant que Bibliothèque Système, n'a pas besoin d'être incluse dans l'acheminement de l'œuvre sous forme de code objet.

Un "Produit Utilisateur" est soit (1) un "Produit de Consommation," ce qui signifie toute propriété personnelle tangible normalement utilisée à des fins personnelles, familiales ou relatives au foyer, soit (2) toute chose conçue ou vendue pour l'incorporation dans un lieu d'habitation. Pour déterminer si un produit constitue un Produit de Consommation, les cas ambigus sont résolus en fonction de la couverture. Pour un produit particulier reçu par un utilisateur particulier, l'expression "normalement utilisée" ci-avant se réfère à une utilisation typique ou l'usage commun de produits de même catégorie, indépendamment du statut de cet utilisateur particulier ou de la façon spécifique dont cet utilisateur particulier utilise effectivement ou s'attend lui-même ou est attendu à utiliser ce produit. Un produit est un Produit de Consommation indépendamment du fait que ce produit a ou n'a pas d'utilisations substantielles commerciales, industrielles ou hors Consommation, à moins que de telles utilisations représentent le seul mode significatif d'utilisation du produit.

Les "Informations d'Installation" d'un Produit Utilisateur signifient toutes les méthodes, procédures, clés d'autorisation ou autres informations requises pour installer et exécuter des versions modifiées d'une Œuvre Couverte dans ce Produit Utilisateur à partir d'une version modifiée de son Source Correspondant. Les informations qui suffisent à assurer la continuité de fonctionnement du code objet modifié ne doivent en aucun cas être empêchées ou interférées du seul fait qu'une modification a été effectuée.

Si vous acheminez le code objet d'une Œuvre Couverte dans, ou avec, ou spécifiquement pour l'utilisation dans, un Produit Utilisateur et si l'acheminement se produit en tant qu'élément d'une transaction dans laquelle le droit de possession et d'utilisation du Produit Utilisateur est transféré au Destinataire définitivement ou pour un terme fixé (indépendamment de la façon dont la transaction est caractérisée), le Source Correspondant acheminé selon cet article-ci doit être accompagné des Informations d'Installation. Mais cette obligation ne s'applique pas si ni vous ni aucune tierce partie ne détient la possibilité d'installer un code objet modifié sur le Produit Utilisateur (par exemple, l'œuvre a été installée en mémoire morte).

L'obligation de fournir les Informations d'Installation n'inclue pas celle de continuer à fournir un service de support, une garantie ou des mises à jour pour une œuvre qui a été modifiée ou installée par le Destinataire, ou pour le Produit Utilisateur dans lequel elle a été modifiée ou installée. L'accès à un réseau peut être rejeté quand la modification elle-même affecte matériellement et défavorablement les opérations du réseau ou viole les règles et protocoles de communication au travers du réseau.

Le Source Correspondant acheminé et les Informations d'Installation fournies, en accord avec cet article, doivent être dans un format publiquement documenté (et dont une implémentation est disponible auprès du public sous forme de code source) et ne doit nécessiter aucune clé ou mot de passe spécial pour le dépaquetage, la lecture ou la copie.

Article 7. Termes additionnels.

Les « permissions additionnelles » désignent les termes qui supplémentent ceux de cette Licence en émettant des exceptions à l'une ou plusieurs de ses conditions. Les permissions additionnelles qui sont applicables au Programme entier doivent être traitées comme si elles étaient incluent dans cette Licence, dans les limites de leur validité suivant la loi applicable. Si des permissions additionnelles s'appliquent seulement à une partie du Programme, cette partie peut être utilisée séparément suivant ces permissions, mais le Programme tout entier reste gouverné par cette Licence sans regard aux permissions additionnelles.

Quand vous acheminez une copie d'une Œuvre Couverte, vous pouvez à votre convenance ôter toute permission additionnelle de cette copie, ou de n'importe quelle partie de celui-ci. (Des permissions additionnelles peuvent être rédigées de façon à requérir leur propre suppression dans certains cas où vous modifiez l'œuvre.) Vous pouvez placer les permissions additionnelles sur le matériel acheminé, ajoutées par vous à une Œuvre Couverte pour laquelle vous avez ou pouvez donner les permissions de Droit d'Auteur ("copyright") appropriées.

Nonobstant toute autre clause de cette Licence, pour tout constituant que vous ajoutez à une Œuvre Couverte, vous pouvez (si autorisé par les titulaires de Droit d'Auteur pour ce constituant) supplémenter les termes de cette Licence avec des termes :

- a) qui rejettent la garantie ou limitent la responsabilité de façon différente des termes des articles 15 et 16 de cette Licence; ou
- b) qui requièrent la préservation de notices légales raisonnables spécifiées ou les attributions d'auteur dans ce constituant ou dans les Notices Légales Appropriées affichées par les œuvres qui le contiennent; ou
- c) qui prohibent la représentation incorrecte de l'origine de ce constituant, ou qui requièrent que les versions modifiées d'un tel constituant soient marquées par des moyens raisonnables comme différentes de la version originale; ou
- d) qui limitent l'usage à but publicitaire des noms des concédants de licence et des auteurs du constituant; ou
- e) qui refusent à accorder des droits selon la législation relative aux marques commerciales, pour l'utilisation dans des noms commerciaux, marques commerciales ou marques de services; ou
- f) qui requièrent l'indemnisation des concédants de licences et auteurs du constituant par quiconque achemine ce constituant (ou des versions modifiées de celui-ci) en assumant contractuellement la responsabilité envers le Destinataire, pour toute responsabilité que ces engagements contractuels imposent directement à ces octroyants de licences et auteurs.

Tous les autres termes additionnels non permissifs sont considérés comme des « restrictions avancées » dans le sens de l'article 10. Si le Programme tel que vous l'avez reçu, ou toute partie de celui-ci, contient une notice établissant qu'il est gouverné par cette Licence en même temps qu'un terme qui est une restriction avancée, vous pouvez ôter ce terme. Si un document de licence contient une restriction avancée mais permet la reconcession de licence ou l'acheminement suivant cette Licence, vous pouvez ajouter une Œuvre Couverte constituante gouvernée par les termes de ce document de licence, pourvu que la restriction avancée ne survit pas à une telle cession de licence ou un tel acheminement.

Si vous ajoutez des termes à une Œuvre Couverte en accord avec cet article, vous devez placer, dans les fichiers sources appropriés, une déclaration des termes additionnels qui s'appliquent à ces fichiers, ou une notice indiquant où trouver les termes applicables.

Les termes additionnels, qu'ils soient permissifs ou non permissifs, peuvent être établis sous la forme d'une licence écrite séparément, ou établis comme des exceptions; les obligations ci-dessus s'appliquent dans chacun de ces cas.

Article 8. Terminaison.

Vous ne pouvez ni propager ni modifier une Œuvre Couverte autrement que suivant les termes de cette Licence. Toute autre tentative de le propager ou le modifier est nulle et terminera automatiquement vos droits selon cette Licence (y compris toute licence de brevet accordée selon le troisième paragraphe de l'article 11).

Cependant, si vous cessez toute violation de cette Licence, alors votre licence depuis un titulaire de Droit d'Auteur ("copyright") est réinstaurée (a) à titre provisoire à moins que et jusqu'à ce que le titulaire de Droit d'Auteur termine finalement et explicitement votre licence, et (b) de façon permanente si le titulaire de Droit d'Auteur ne parvient pas à vous notifier de la violation par quelque moyen raisonnable dans les soixante (60) jours après la cessation.

De plus, votre licence depuis un titulaire particulier de Droit d'Auteur est réinstaurée de façon permanente si ce titulaire vous a notifié de la violation par quelque moyen raisonnable, et si c'est la première fois que vous avez reçu une notification de violation de cette Licence (pour une œuvre quelconque) depuis ce titulaire de Droit d'Auteur, et si vous résolvez la violation dans les trente (30) jours qui suivent votre réception de la notification.

La terminaison de vos droits suivant cette section ne terminera pas les licences des parties qui ont reçu des copies ou droits de votre part suivant cette Licence. Si vos droits ont été terminés et non réinstaurés de façon permanente, vous n'êtes plus qualifié à recevoir de nouvelles licences pour les mêmes constituants selon l'article 10.

Article 9. Acceptation non requise pour obtenir des copies.

Vous n'êtes pas obligé d'accepter cette licence afin de recevoir ou exécuter une copie du Programme. La propagation asservie d'une Œuvre Couverte qui se produit simplement en conséquence d'une transmission d'égal-à-égal pour recevoir une copie ne nécessite pas l'acceptation. Cependant, rien d'autre que cette Licence ne vous accorde la permission de propager ou modifier une quelconque Œuvre Couverte. Ces actions enfreignent le Droit d'Auteur si vous n'acceptez pas cette Licence. Par conséquent, en modifiant ou propageant une Œuvre Couverte, vous indiquez votre acceptation de cette Licence pour agir ainsi.

Article 10. Cession automatique de Licence aux Destinataires et intermédiaires.

Chaque fois que vous acheminez une Œuvre Couverte, le Destinataire reçoit automatiquement une licence de la part des concédants originaux, pour exécuter, modifier et propager cette œuvre, suivant les termes de cette Licence. Vous n'êtes pas responsable du renforcement de la conformation des tierces parties aux termes de cette Licence.

Une "transaction d'entité" désigne une transaction qui transfère le contrôle d'une organisation, ou de substantiellement tous ses actifs, ou la subdivision d'une organisation, ou la fusion de plusieurs organisations. Si la propagation d'une Œuvre Couverte résulte d'une transaction d'entité, chaque partie à cette transaction qui reçoit une copie de l'œuvre reçoit aussi les licences pour l'œuvre que le prédécesseur intéressé à cette partie avait ou pourrait donner selon le paragraphe précédent, plus un droit de possession du Source Correspondant de cette œuvre depuis le prédécesseur intéressé si ce prédécesseur en dispose ou peut l'obtenir par des efforts raisonnables.

Vous ne pouvez imposer aucune restriction avancée dans l'exercice des droits accordés ou affirmés selon cette Licence. Par exemple, vous ne pouvez imposer aucun paiement pour la licence, aucune royaltie, ni aucune autre charge pour l'exercice des droits accordés selon cette Licence; et vous ne pouvez amorcer aucun litige judiciaire (y compris une réclamation croisée ou contre-réclamation dans un procès) sur l'allégation qu'une revendication de brevet est enfreinte par la réalisation, l'utilisation, la vente, l'offre de vente, ou l'importation du Programme ou d'une quelconque portion de celui-ci.

Article 11. Brevets.

Un « contributeur » est un titulaire de Droit d'Auteur (**"copyright"**) qui autorise l'utilisation selon cette Licence du Programme ou de l'œuvre sur laquelle le Programme est basé. L'œuvre ainsi soumise à licence est appelée la "version contributive" de ce contributeur.

Les "revendications de brevet essentielles" sont toutes les revendications de brevets détenues ou contrôlées par le contributeur, qu'elles soient déjà acquises par lui ou acquises subséquemment, qui pourraient être enfreintes de quelque manière, permises par cette Licence, sur la réalisation, l'utilisation ou la vente de la version contributive de celui-ci. Aux fins de cette définition, le "contrôle" inclue le droit de concéder des sous-licences de brevets d'une manière consistante, nécessaire et suffisante, avec les obligations de cette Licence.

Chaque contributeur vous accorde une licence de brevet non exclusive, mondiale et libre de toute royaltie, selon les revendications de brevet essentielles, pour réaliser, utiliser, vendre, offrir à la vente, importer et autrement exécuter, modifier et propager les contenus de sa version contributive.

Dans les trois paragraphes suivants, une "licence de brevet" désigne tous les accords ou engagements exprimés, quel que soit le nom que vous lui donnez, de ne pas mettre en vigueur un brevet (telle qu'une permission explicite pour mettre en pratique un brevet, ou un accord pour ne pas poursuivre un Destinataire pour cause de violation de brevet). "Accorder" une telle licence de brevet à une partie signifie conclure un tel accord ou engagement à ne pas faire appliquer le brevet à cette partie.

Si vous acheminez une Œuvre Couverte, dépendant en connaissance d'une licence de brevet, et si le Source Correspondant de l'œuvre n'est pas disponible à quiconque copie, sans frais et suivant les termes de cette Licence, à travers un serveur réseau publiquement accessible ou tout autre moyen immédiatement accessible, alors vous devez soit (1) rendre la Source Correspondante ainsi disponible, soit (2) vous engager à vous priver pour vous-même du bénéfice de la licence de brevet pour cette œuvre particulière, soit (3) vous engager, d'une façon consistante avec les obligations de cette Licence, à étendre la licence de brevet aux Destinataires de cette œuvre. "Dépendant en connaissance" signifie que vous avez effectivement connaissance que, selon la licence de brevet, votre acheminement de l'Œuvre Couverte dans un pays, ou l'utilisation de l'Œuvre Couverte par votre Destinataire dans un pays, enfreindrait un ou plusieurs brevets identifiables dans ce pays où vous avez des raisons de penser qu'ils sont valides.

Si, conformément à ou en liaison avec une même transaction ou un même arrangement, vous acheminez, ou propagez en procurant un acheminement de, une Œuvre Couverte et si accordez une licence de brevet à l'une des parties recevant l'Œuvre Couverte pour lui permettre d'utiliser, propager, modifier ou acheminer une copie spécifique de l'Œuvre Couverte, alors votre accord est automatiquement étendu à tous les Destinataires de l'Œuvre Couverte et des œuvres basées sur celle-ci.

Une licence de brevet est "discriminatoire" si, dans le champ de sa couverture, elle n'inclut pas un ou plusieurs des droits qui sont spécifiquement accordés selon cette Licence, ou en prohibe l'exercice, ou est conditionnée par le non-exercice d'un ou plusieurs de ces droits. Vous ne pouvez pas acheminer une Œuvre Couverte si vous êtes partie à un arrangement, selon lequel une partie tierce exerçant son activité dans la distribution de logiciels et à laquelle vous effectuez un paiement fondé sur l'étendue de votre activité d'acheminement de l'œuvre, et selon lequel la partie tierce accorde, à une quelconque partie qui recevrait depuis vous l'Œuvre Couverte, une licence de brevet discriminatoire (a) en relation avec les copies de l'Œuvre Couverte acheminées par vous (ou les copies réalisées à partir de ces copies), ou (b) avant tout destinée à et en relation avec des produits spécifiques ou compilations contenant l'Œuvre Couverte, à moins que vous ayez conclu cet arrangement ou que la licence de brevet ait été accordée avant le 28 mars 2007.

Rien dans cette Licence ne devrait être interprété comme devant exclure ou limiter toute licence implicite ou d'autres moyens de défense à une infraction qui vous seraient autrement disponible selon la loi applicable relative aux brevets.

Article 12. Non abandon de la liberté des autres.

Si des conditions vous sont imposées (que ce soit par décision judiciaire, par un accord ou autrement) qui contredisent les conditions de cette Licence, elles ne vous excusent pas des conditions de cette Licence. Si vous ne pouvez pas acheminer une Œuvre Couverte de façon à satisfaire simultanément vos obligations suivant cette Licence et toutes autres obligations pertinentes, alors en conséquence vous ne pouvez pas du tout l'acheminer. Par exemple, si vous avez un accord sur des termes qui vous obligent à collecter pour le réacheminement des royalties depuis ceux à qui vous acheminez le Programme, la seule façon qui puisse vous permettre de satisfaire à la fois à ces termes et ceux de cette Licence sera de vous abstenir entièrement d'acheminer le Programme.

Article 13. Utilisation avec la Licence Générale Publique Affero GNU.

Nonobstant toute autre clause de cette Licence, vous avez la permission de lier ou combiner toute Œuvre Couverte avec une œuvre placée sous la version 3 de la Licence Générale Publique GNU Affero ("GNU Affero General Public License") en une seule œuvre combinée, et d'acheminer l'œuvre résultante. Les termes de cette Licence continueront à s'appliquer à la partie formant une Œuvre Couverte, mais les obligations spéciales de la Licence Générale Publique GNU Affero, article 13, concernant l'interaction à travers un réseau, s'appliqueront à la combinaison en tant que telle.

Article 14. Versions révisées de cette Licence.

La Free Software Foundation peut publier des versions révisées et/ou nouvelles de la Licence Publique Générale GNU ("GNU General Public License") de temps en temps. De telles version nouvelles resteront similaires dans l'esprit avec la présente version, mais peuvent différer dans le détail afin de traiter de nouveaux problèmes ou préoccupations.

Chaque version reçoit un numéro de version distinctif. Si le Programme indique qu'une version spécifique de la Licence Publique Générale GNU "ou toute version ultérieure" ("or any later version") s'applique à celui-ci, vous avez le choix de suivre soit les termes et conditions de cette version numérotée, soit ceux de n'importe quelle version publiée ultérieurement par la Free Software Foundation. Si le Programme n'indique pas une version spécifique de la Licence Publique Générale GNU, vous pouvez choisir l'une quelconque des versions qui ont été publiées par la Free Software Foundation.

Si le Programme spécifie qu'un intermédiaire peut décider quelles versions futures de la Licence Générale Publique GNU peut être utilisée, la déclaration publique d'acceptation d'une version par cet intermédiaire vous autorise à choisir cette version pour le Programme.

Des versions ultérieures de la licence peuvent vous donner des permissions additionnelles ou différentes. Cependant aucune obligation additionnelle n'est imposée à l'un des auteurs ou titulaires de Droit d'Auteur du fait de votre choix de suivre une version ultérieure.

Article 15. Déclaration d'absence de garantie.

Il n'y a aucune garantie pour le programme, dans les limites permises par la loi applicable. À moins que cela ne soit établi différemment par écrit, les propriétaires de droits et/ou les autres parties fournissent le programme "en l'état" sans garantie d'aucune sorte, qu'elle soit exprimée ou implicite, ceci comprenant, sans se limiter à celles-ci, les garanties implicites de commercialisabilité et d'adéquation à un objectif particulier. Vous assumez le risque entier concernant la qualité et les performances du programme. Dans l'éventualité où le programme s'avérerait défectueux, vous assumez les coûts de tous les services, réparations ou corrections nécessaires.

Article 16. Limitation de responsabilité.

En aucune autre circonstance que celles requises par la loi applicable ou accordées par écrit, un titulaire de droits sur le programme, ou tout autre partie qui modifie ou achemine le programme comme permis ci-dessus, ne peut être tenu pour responsable envers vous pour les dommages, incluant tout dommage général, spécial, accidentel ou induit survenant par suite de l'utilisation ou de l'incapacité d'utiliser le programme (y compris, sans se limiter à celles-ci, la perte de données ou l'inexactitude des données retournées ou les pertes subies par vous ou des parties tierces ou l'incapacité du programme à fonctionner avec tout autre programme), même si un tel titulaire ou toute autre partie a été avisé de la possibilité de tels dommages.

Article 17. Interprétation des sections 15 et 16.

Si la déclaration d'absence de garantie et la limitation de responsabilité fournies ci-dessus ne peuvent prendre effet localement selon leurs termes, les cours de justice qui les examinent doivent appliquer la législation locale qui approche au plus près possible une levée absolue de toute responsabilité civile liée au Programme, à moins qu'une garantie ou assumation de responsabilité accompagne une copie du Programme en échange d'un paiement.