# HOWTO Java pour Linux

maintenu par Eric S. Raymond (esr@snark.thyrsus.com)
Adaptation française par Arnaud Ruch (ruch@donald.fdn.fr) v.2.2, 10 Fev 1996

Ce document décrit de façon succinte la mise en place et l'utilisation de Java et HotJava sous Linux. Il s'adresse aussi bien aux utilisateurs qu'aux programmeurs.

## Table des matières

# 1 Introduction

Ce document est une introduction Linux-centrique aux mondes de Java et HotJava. Ces technologies connaissant une évolution rapide, toute contribution est la bienvenue.

Certaines parties de ce document demandent sérieusement à être vérifiées ou complétées [cf. 9]. Pour aider à l'amélioration de ce HOWTO, envoyez toute demande de mise à jour et de modification à esr@snark.thyrsus.com.

# 2 Questions d'ordre général sur Java et HotJava

Cette partie est une introduction générale (non spécifique à Linux) à Java et HotJava.

## 2.1 Que sont Java et HotJava? En quoi sont-ils intéressants?

Java est un langage orienté réseau. Similaire au C++, il est plus compact et a été conçu plus proprement. Il met en oeuvre une gestion mémoire semblable à celle du Lisp (ramasse-miettes), mais dispose en plus d'une vérification statique des types, raison pour laquelle il est parfois décrit avec ironie comme un « Smalltalk doté d'une syntaxe cohérente ». Les processus légers (threads) y sont intégrés et il dispose de nombreuses possibilités en termes de sécurité réseau. Jusqu'à présent, la principale application réalisée grâce à Java est le navigateur WWW HotJava, mais l'avenir de Java comme langage de développement d'applications de tous types semble prometteur.

HotJava est un navigateur WWW écrit en Java. Son principal avantage sur ses concurrents est de pouvoir traiter une nouvelle construction HTML appelée « APPLET ». Une applet est un programme Java qui s'exécute sur la machine cliente. Les documents WWW peuvent par conséquent contenir des objets « vivants » et non plus de simples données.

La possibilité de transférer des objets exécutables représente le progrès le plus significatif de la technologie du World Wide Web depuis l'apparition de Mosaic. L'application la plus immédiate en est de pouvoir concevoir un navigateur Web qui ne nécessite plus d'incessantes mises à jour pour être capable de traiter les nouveaux formats d'images et les nouvelles constructions.

Un programme Java est transformé par compilation en un pseudo-code appelé « J-code » que l'interpréteur Java a pour rôle d'exécuter. Il se pourrait bien que le J-code soit en passe de devenir une référence universelle en matière de code indépendant s'exécutant sur une machine virtuelle, ou encore comme langage de compilation intermédiaire. Des rumeurs courent à propos de l'existence d'un compilateur ADA-95 générant du J-code et sur le développement d'un tel compilateur pour le langage C. Il est probable qu'en fin de compte, de nombreux langages de haut niveau permettront d'écrire des applets ou des programmes complets sur toute machine disposant d'un interpréteur Java. On peut aussi signaler l'apparition de microprocesseurs Java dont le jeu d'instruction est le J-code.

## 2.2 D'où viennent Java et HotJava? Qui a le droit de les utiliser?

Java et HotJava ont été développés chez Sun Microsystems, par une équipe conduite par James Gosling (déjà largement connu pour Gosling Emacs et NeWS).

La dernière tentative menée par Sun pour imposer un standard technique majeur a été NeWS, son système de fenêtrage réseau (Network Window System). Bien que la supériorité de NeWS sur X ait été reconnue presque unanimement, X a gagné la bataille pour la simple raison que les sources en étaient gratuitement disponibles. Cet échec a été formateur pour Sun : Java et HotJava sont aisément accessibles, les fichiers source peuvent être téléchargés selon les termes d'une licence relativement souple [cf. 3] et Sun encourage les portages vers des environnements autres que Sun.

Netscape est déjà capable d'interpréter du Java. Microsoft a acquis la licence de la technologie en Décembre 1995. Il semble donc que le support de Java soit en passe de devenir universel en 1996.

A l'origine, Java s'appelait Oak. HotJava fut quant à lui dénommé WebRunner.

## 2.3 Quelle est la maturité de Java?

Le 12 Décembre 1995, Sun mettait à disposition du public la version 'Bêta 2' de Java. Un mois plus tard venait le tour de la version 1.0, première version dite de « production ». L'API (Application Programmers Interface) de l'environnement Java, qui définit son accès au système d'exploitation et au système de fenêtrage de la machine hôte, semble avoir été « gelée » : elle pourra être étendue, mais ne sera pas rendue incompatible avec la version actuelle.

La version 1.0 de ce document a été rendue publique en même temps que la version bêta 1.0 du kit de développement Java (JDK) pour Linux, dont le portage a été réalisé par Randy Chapman.

C'est un fait connu que de sérieux trous de sécurité existent dans l'implémentation de Java. Ce n'est donc pour l'instant pas une très bonne idée que d'utiliser Java pour des applications sensibles. Ces problèmes seront certainement résolus dans les versions de production (post-beta).

## 2.4 Où peut-on trouver de la documentation sur Java et HotJava?

Sun tient à jour un serveur Web complet proposant des documents sur Java et HotJava à l'adresse : http://java.sun.com.
Des copies de ces documents se trouvent en d'autres endroits, tels que http://java.blackdown.com (Sun fournit une liste de sites miroirs).

## 2.5 Certes, mais où trouver une documentation papier ?

La série officielle des manuels Java de SunSoft Press est en cours de production chez Addison-Wesley. On peut trouver quelques détails sur ces ouvrages dans comp.lang.java FAQ .

Certains des pionniers de Java ont recommandé l'ouvrage « Hooked on Java », écrit par les membres de l'équipe de développement de Java. L'un des commentaires qui ont été faits sur ce livre est : « Bien écrit, mais j'aurais apprécié quelque chose de moins basique ».

O'Reilly & Associates travaille actuellement sur sa propre série d'ouvrages traitant de Java, en collaboration avec Sun. On trouvera (au moins) un « Nutshell Guide to Java », un manuel de référence du langage, un manuel de référence des bibliothèques de classes et un ouvrage traitant de la machine virtuelle sous-jacente à Java (celle qui exécute le J-code).

Information tout à fait non confidentielle : l'auteur de ce HOWTO a été prié par O'Reilly & Associates de servir de principal relecteur technique de ces ouvrages et il est rémunéré pour ce travail.

SamsNet a publié un livre intitulé « Learning Java in 21 Days ». Aucun commentaire n'en a été fait pour l'instant.

Un des contributeurs de ce document a trouvé dans des librairies proches de chez lui trois ouvrages déjà publiés :

- L'un, appelé « Java in 60 minutes » semble n'être qu'une simple description syntaxique du langage (environ $20) ;

- Le second, publié par SamsNet, s'appelle tout simplement « Java » (environ $20) ;
- Le troisième, écrit par Tim Ritchey, est intitulé « Java! ». Un CD-ROM est fourni avec (environ $35).

Notre informateur poursuit : « J'ai acheté le troisième (écrit par Tim Ritchey). Les délais d'impressions étant ce qu'ils sont, le CD-ROM ne contient que des versions alpha, bien qu'il signale que la version bêta est probablement disponible au moment de la publication, et que les informations les plus à jour se trouvent sur java.sun.com. Cela mis à part, c'est un bon ouvrage et il juge même nécessaire de signaler qu'il est inutile de s'embarrasser de centaines de pages de description de l'API parce que : cela quadruplerait la taille du livre, l'API est encore en évolution, et ces informations se trouvent on-line... »

Dave Dittrich indique : « Je dirais que « Java in 60 minutes » peut certes être lu en 60 minutes, mais qu'il ne vous apprendra certainement pas en 60 minutes comment programmer en Java (surprise, surprise !). J'ai l'impression que « Java in 60 minutes » est l'une de ces nombreuses tentatives de publication d'un ouvrage sur un sujet à la pointe de l'actualité. On n'y trouve apparemment pas grand'chose de plus que ce qui est disponible sur le Web et une grande quantité de texte donne l'impression d'avoir été reproduite telle quelle à partir du document « Java Language Specification » http://java.sun.com/JDK-beta2/psfiles/javaspec.ps. »

Dave poursuit : « Il en va de même avec « Java! », mais ce dernier détaille beaucoup plus les concepts de la programmation orientée objet. Quelqu'un m'a signalé un autre ouvrage qui traite assez bien de l'abstraction données/méthodes et des autres concepts de la programmation orientée objet. Il s'agit de "Structure and Interpretation of Computer Programs", par Abelson, Sussman et Sussman (MIT Press). »

Dernière mise en garde : à l'heure actuelle (Janvier 1996), c'est probablement une bonne idée que d'éviter le livre de Ritchey ou tout ce qui peut être basé sur l'API alpha. L'API bêta est substantiellement différente et Java 1.0, qui devrait bientôt être disponible, aura encore ses propres différences.

## 2.6 Où peut-on trouver des collections de code Java disponibles on-line ?

On trouve des archives d'applets sur le WWW, aux adresses suivantes :

- http://java.sun.com/ ;
- http://www.applets.com/ ;
- http://www.gamelan.com/ ;
- http://www.javasoft.com/applets/applets.html.

La plupart des applets présentées sur ces pages sont accompagnées de leur code source et les programmeurs sont encouragés à les utiliser.

Des pointeurs vers d'autres ressources peuvent être trouvés dans le comp.lang.java FAQ.

# 3 Java, logiciel librement diffusable (Freely Redistributable Software)

Nombreux sont les programmeurs Linux qui tiennent à ce que les logiciels qu'ils produisent soient librement diffusables et qui évitent de consacrer trop de temps aux outils dont les sources ne sont pas largement disponibles. Dans cette partie sont discutées les relations de Sun et de Java avec le monde du logiciel librement diffusable.

(Attention : l'auteur n'est pas juriste. Il n'est pas non plus employé, agent ni actionnaire de Sun Microsystems. Cette partie est fondée sur son interprétation des lois actuelles sur le copyright et sur la licence de Sun. Considérez-la comme une simple introduction, la page de Sun http://www.blackdown.org/Java/licensing.html étant la seule autorité en la matière. Si vous avez du mal à en saisir la signification, consultez un avocat.)

## 3.1 Peut-on redistribuer librement des programmes écrits en Java ?

Tout à fait, si vous en êtes l'auteur et choisissez de les placer sous une licence de type GPL ou BSD, ou encore si vous les déclarez du domaine public.

## 3.2 Quelles sont les droits et obligations associés aux exécutables de Java ?

La page de Sun concernant la licence indique que les exécutables de Java et HotJava peuvent être redistribués dans des applications (commerciales ou non), à condition qu'ils le soient gratuitement et sans être modifiés. Les entreprises ou individus qui souhaitent effectuer un nouveau portage du langage Java peuvent mettre les exécutables de ce portage à disposition sur l'Internet, mais ceux-ci doivent être gratuits et passer avec succès les tests publiquement disponibles (ces tests devraient être disponibles à la mi-1996). Les fichiers « diff » peuvent également être diffusés, à condition que les sources auxquels ils s'appliquent ne le soient pas.

Ainsi, vous pouvez donner à vos amis les exécutables de Java et HotJava ou bien les mettre sur un CD-ROM, ou encore en faire ce que vous voudrez, sauf les modifier et en diffuser les versions modifiées comme étant Java (ceci est compréhensible, Sun doit se protéger contre les chevaux de Troie diffusés sous le déguisement de Java). Les exécutables peuvent être redistribués librement.

## 3.3 Quelles parties des sources de Java peut-on redistribuer librement ?

Selon la page de Sun, aucune. Cependant, les choses sont en réalité moins strictes que cela. En pratique, tout le monde peut obtenir les sources de Java et HotJava à des fins non commerciales, d'enseignement ou de portage, et ce en remplissant sur le Web un formulaire qui en interdit la redistribution. De plus, il est permis de redistribuer des fichiers "diff" à appliquer aux sources.

Le seul cas où il faut signer un accord commercial avec Sun et leur verser de l'argent est celui de l'utilisation des sources à des fins commerciales. Cependant, il est dans l'esprit de cet accord de signer la licence d'utilisation non-commerciale des sources, de les récupérer, de les porter et de distribuer totalement librement les exécutables résultants !

Sun affirme que le but principal d'un tel contrôle est d'empêcher le langage de muter vers des dialectes incompatibles. C'est ce vers quoi tend la licence. Elle semble cependant avoir été conçue de façon à permettre aux hackers de s'amuser gratuitement.

## 3.4 Quelle est l'attitude de Sun vis-à-vis des implémentations et outils Java gratuits ?

D'après la page de licence de Sun :

> Les spécifications du langage Java et de la machine virtuelle Java sont OUVERTES et sous copyright de Sun Microsystems, Inc.
>
> Les nouvelles implémentations du compilateur et de l'interpréteur Java sont autorisées sans avoir besoin de licence de Sun, du moment que de telles implémentations sont créées à partir des spécifications publiées et sans utilisation directe ou indirecte d'implémentations, de propriétés intellectuelles ou marques commerciales de Sun.

(NdT la traduction ci-dessus n'engage ni Sun ni l'auteur de ce document) Ainsi, si vous voulez vous lancer dans la création d'un clone gratuit de Java à partir des spécifications publiées, Sun ne fera rien pour vous en empêcher.

Les gens de Sun utilisent le terme de « portage sauvage » pour désigner les implémentations de Java qui sont réalisées ailleurs que chez Sun ou ne s'appuient pas sur du code dont Sun detient la licence. Parfois le terme semble avoir l'une de ces significations, parfois l'autre. Ils ont l'air de penser que de telles implémentations existent déjà, mais nous ne savons pas où l'on peut les trouver, ni si elles sont gratuites. Les gens de Sun semblent envisager cette situation avec sérénité.

Sun a mis au point un ensemble de logiciels de validation. Sun affirme que sera certifié tout portage Java qui passe ces tests avec succès. Cela sera-t-il gratuit ? Nous n'en savons rien, mais Sun affirme que ces logiciels seront « disponibles publiquement » en 1996, ce qui semble orienter vers la gratuité. Ils sont également prêts à certifier des « portages sauvages » ce qui, apparemment, n'est pas encore arrivé.

## 3.5 Peut-on trouver des clones de Java dont les sources soient librement redistribuables ?

On peut trouver un clone du compilateur Java en début de développement. Il s'appelle guavac et vous pourrez trouver des informations à son sujet à l'adresse http://http.cs.berkeley.edu/ engberg/guavac.

Des rumeurs ont couru indiquant que le projet GROW de Cygnus Software prévoyait d'intégrer un interpréteur de code objet Java. Elles étaient fausses : ce qui est en cours de développement est un traducteur de Java vers Scheme qui ne passe pas par le J-code. Pour plus de détails sur le projet GROW, consultez http://www.cygnus.com/tiemann/grow/.

Erik Troan de Red Hat et l'auteur de ce document tentent de former un groupe de développement pour mener à bien le reste du travail. Actuellement, l'idée est d'utiliser guavac comme compilateur, Erik Troan s'occupant de la bibliothèque de classes et l'auteur de ce HOWTO de l'interpréteur de J-code (si Cygnus n'y parvient pas auparavant). Ce projet devait être nommé « Mr Coffee », ce qui était tentant, mais s'appellera probablement JOE (Java Open to Everyone).

### 3.6 Peut-on trouver des outils Java librement redistribuables ?

Vous pourrez trouver un mode GNU Emacs pour l'édition de Java à l'adresse
ftp://java.sun.com/pub/java/contrib/emacs.
Note : ce mode requiert l'utilisation de `c++-mode.el` et ne fonctionne pas avec le mode `cc-mode.el` distribué avec Emacs 19.

## 4 Questions concernant Java sous Linux

Vous apprendrez ici les astuces de base permettant d'utiliser Java sous Linux.

### 4.1 Peut-on trouver Java et HotJava dans une distribution Linux ?

Pas pour l'instant, mais cela ne devrait être qu'une question de semaines.

### 4.2 Où trouver la dernière distribution de Java pour Linux ?

Jetez un coup d'oeil à ftp://java.blackdown.org/pub/Java/linux. Le dernier Java pour Linux peut y être récupéré. Il vaut mieux cependant commencer par lire ftp://java.blackdown.org/pub/Java/linux/README.

Les fichiers dont vous aurez besoin sont :

```
             linux.jdk-1.0-try1.common.tar.gz
     et soit :
             linux.jdk-1.0-try3.static-motif-bin.tar.gz
        soit :
             linux.jdk-1.0-try3.shared-motif-bin.tar.gz,
```

suivant la présence ou non des bibliothèques partagées Motif sur votre système.

Si vous utilisez Red Hat Linux ou tout autre système connaissant RPM, vous pouvez installer la beta 1.0 à partir de :
ftp://java.blackdown.org/pub/Linux/RPMS/i386/JDK-linux-common-1.0b-1.i386.rpm
et
ftp://java.blackdown.org/pub/Linux/RPMS/i386/JDK-linux-static-1.0b-1.i386.rpm
ou
ftp://java.blackdown.org/pub/Linux/RPMS/i386/JDK-linux-shared-1.0b-1.i386.rpm
.

Pour les utilisateurs Européens, le contenu de ftp://java.blackdown.org/pub/Java/linux et
ftp://java.blackdown.org/pub/Linux/RPMS/i386/JDK-linux* est désormais répliqué en Grande-Bretagne à

l'adresse : ftp://ftp.linux.org.uk/pub/linux/java. Si cela intéresse quelqu'un, il s'agit de la machine `zen.btc.uwe.ac.uk`. Le miroir est réactualisé tous les jours vers 6 heures du matin, heure locale.

Ce portage (réalisé par Randy Chapman à partir des sources de Sun) est celui auquel il est fait référence partout ailleurs dans ce document sous le nom de JDK (Java Development Kit).

## 4.3 Quel est l'environnement requis pour exécuter Java?

Pour commencer, il faut un système Linux ELF. Aucun support a.out n'est disponible et, vue la vitesse à laquelle le monde Linux passe à ELF, il est peu probable qu'il y en ait jamais un.

Il vous faut un noyau 1.2.13 ou plus récent. Il semble y avoir un bug dans la fonction `getcwd(3)` des noyaux antérieurs, ce qui est fatal à bin/javac qui n'en vérifie pas le code de retour.

Ces bibliothèques sont également nécessaires :

**libc.so.5 => /lib/libc.so.5.2.18**
: (La bibliothèque C standard)

**libX11.so.6 => /usr/X11/lib/libX11.so.6.0**
: (La bibliothèque de base de X11R6)

**libXt.so.6 => /usr/X11/lib/libXt.so.6.0**
: (La bibliothèque de la boîte à outils Athena)

**libXext.so.6 => /usr/X11/lib/libXext.so.6.0**
: (La bibliothèque d'extensions X)

**libXpm.so.4 => /usr/X11/lib/libXpm.so.4.3**
: (La bibliothèque X de gestion des pixmaps)

**libdl.so.1 => /lib/libdl.so.1.7.9**
: (Le support du chargement dynamique de modules objet)

Il n'est pas impossible que les bibliothèques de support du C et de X soient déjà installées sur votre système.

Si vous ne l'avez pas déjà, récupérez `libc.5.2.18.bin.tar.gz` sur ftp://tsx-11.mit.edu/pub/linux/packages/GCC/. Décompressez et détarez-le, puis copiez dans /lib les fichiers contenus dans le répertoire lib créé.

Si vous n'avez pas une version récente de `ld.so` (ou `libdl.so.1.7.*`), il vous faut la mettre à jour. `ld.so.1.7.9` et `ld.so.1.7.11` fonctionnent, mais pas les versions plus anciennes ni `ld.so.1.7.10`. Récupérez `ld-so.1.7.14.tar.gz` sur ftp://ftp.ods.com/linux/.
Décompressez-le et détarez-le, puis lancez *instldso.sh* qui se trouve à la racine de l'arborescence ainsi créée.

Vous trouverez une bibliothèque Xpm adéquate sur ftp://sunsite.unc.edu/pub/Linux/libs/X/libXpm.3.4f-ELF.tar.gz.

## 4.4 Ai-je besoin de Netscape ? Puis-je l'utiliser ?

Netscape est indispensable, car HotJava n'est pas encore disponible pour les versions beta 1.0. Selon Randy Chapman,

> Sun a fortement modifié l'interface awt dans le JDK et HotJava n'a pas encore été adapté. Sun a promis de le faire et la nouvelle version de HotJava devrait fonctionner parfaitement avec linux-jdk.

Aucune date n'a été annoncée.

Netscape supporte Java depuis la version 2.0b3. On le trouve sur ftp[1-7].netscape.com (i.e. sur ftp1.netscape.com, ftp2.netscape.com,... etc.). La version 2.0b4 permet de lire des applets depuis les URL de type file:, ce que ne permettait pas la version 2.0b3. Ceci devrait être d'un grand secours pour ceux qui ne disposent que de peu de mémoire (il n'est plus nécessaire de faire tourner un démon httpd). La version actuelle est la 2.0b6.

Le fichier à récupérer est : /2.0beta6/unix/netscape-v20b6a-export.i486-unknown-linux.tar.Z.

Notez que Netscape tourne aussi sur des systèmes a.out. Il est donc possible, en théorie, d'écrire une application Java, de la faire compiler par quelqu'un d'autre (comme par exemple BlackStar Public Compiler), et d'en voir le résultat sur votre système. Si cela ne vous suffit pas, passez en ELF.

Assurez-vous que la variable d'environnement CLASSPATH ne soit *pas* positionnée lorsque vous lancez netscape : cela a tendance à provoquer des plantages de celui-ci. Si vous suivez ces conseils, vous ne devriez pas avoir trop de problèmes.

La version Linux de netscape a quelques problèmes connus. D'une part, il n'y a pas de son. D'autre part, seul l'affichage en 8 bits fonctionne : l'affichage en 16 bits provoque des plantages. La version 2.0b3 est parfois sujette à des 'bus errors' lorsqu'elle fait appel à la fonction 'dispose' des frames.

Bien que relativement robuste (cela mis à part), elle ne résiste pas à l'applet 'Impressionism' de Gamelan.

## 4.5 Quels sont les newsgroups et mailing lists traitant de Java sur Linux ?

- *java-linux* : (maintenue by karl@blackdown.org) Discussions et développements concernant le portage de Java sur le système d'exploitation Linux. Envoyez un mail à java-linux-request@java.blackdown.org avec le mot « subscribe » pour être ajouté à la liste ;
- *java-linux-announce* : (maintenue by karl@blackdown.org) Liste modérée des annonces concernant les projets de portage Java-Linux. Envoyez un mail à java-linux-announce-request@java.blackdown.org avec le mot « subscribe » pour être ajouté à la liste ;
- comp.lang.java : Newsgroup traitant de Java en général ;
- alt.www.hotjava : Newsgroup traitant du navigateur HotJava.

# 5 Mise en place de Java sur les principales distributions Linux du marché

Dans cette partie, nous rassemblons les recettes que nous avons reçues concernant la mise en place du portage de Java par Randy Chapman sur les principales distributions de Linux.

## 5.1 Distribution Slackware ELF noyau 1.2.13.

John Franks (john@math.nwu.edu) indique que les étapes suivantes ont été fructueuses :

- Récupérez linux-x86.jdk.static-motif.tar.gz sur ftp://www.blackdown.org/pub/Java/linux/, décompressez-le et détarez-le (les noms des fichiers à récupérer ont changé pour le JDK beta 1.0) ;
- Récupérez libc.5.2.18.bin.tar.gz (la distribution compilée, pas les sources) sur ftp://tsx-11.mit.edu/pub/linux/packages/GCC/. Décompressez-le et détarez-le, puis copiez dans le répertoire /lib les fichiers présents dans le sous-répertoire lib de la racine de l'arborescence créée. Assurez-vous que /lib/libc.so.5 soit un lien symbolique vers ce fichier ;
- Récupérez ld-so.1.7.14.tar.gz sur ftp://ftp.ods.com/linux/. Décompressez-le et détarez-le , puis exécutez "instldso.sh", qui se trouve à la racine de l'arborescence résultante.

Vous devriez maintenant être en mesure de compiler et de tester le programme et l'applet "hello world" de Sun en suivant les indications de http://java.sun.com/progGuide/index.html.

## 5.2 RedHat 2.1/Caldera Preview 2

Steve Greene (sgreene@access.digex.net) rapporte le succès des étapes suivantes. Nous avons quelque peu modifié la recette ; les erreurs sont donc à porter à notre compte. Le JDK auquel il fait allusion est le portage du Java Developer's Kit réalisé par Randy Chapman.

- Récupérez le fichier rpm contenant `ld.so.1.7.14` sur un site miroir de Red Hat. Pour le trouver, cherchez `/pub/mirrors/redhat-2.1/updates/RPMS/ld.so-1.7.14-i386.rpm` dans un site miroir quelconque de Red Hat.

  Je sais qu'on peut le trouver sur ftp.pht.com et ftp.caldera.com. Il se trouve probablement également sur d'autres sites miroir. Récupérez-le et lancez '`rpm -Uvh ld.so-1.7.14-1.i386.rpm`' sur votre machine ;
- Les utilisateurs de Red Hat 2.0 ou de Caldera Preview 2 pourront avoir besoin du programme 'which' (qui est appelé par certains des scripts du JDK). Un fichier rpm pour 'which' est disponible sur les sites miroir habituels de RedHat, comme élément de la distribution Red Hat 2.1. Récupérez et installez `which-1.0.i386.rpm`. Les utilisateurs de RedHat 2.1 doivent normalement déjà avoir which sur leur machine ou disponible dans leur distribution ;
- Récupérez le(s) fichier(s) tar du JDK linké en statique. Détarez-le en un endroit pratique. J'ai placé le mien dans `/usr/local`, si bien qu'est créé `/usr/local/java/...` ;

- Afin de pouvoir exécuter java en tant qu'utilisateur autre que root (ce qui est une bonne idée !), faites un `chmod 666 /dev/zero`. Red Hat et Caldera ont par défaut des droits incorrects sur ce périphérique ;
- Si vous ne l'avez pas déjà fait, lancez ldconfig en tant que root pour charger les nouvelles bibliothèques. Si vous utilisez '`ldconfig -v`', vous pourrez vérifier que toutes les bibliothèques soient bien chargées (au moins, nous n'avons pas à commencer par bâtir un système ELF !) ;
- Assurez-vous d'être connecté en tant qu'utilisateur classique (e.g. pas root). Lancez X-Window, un xterm et essayez quelque chose !

## 5.3 Unifix

Stefan Middenforf écrit : Unifix est une distribution de Linux en langue allemande, assez populaire. Quelques autres distribution s'en servent comme base : Linux Universe, vendue aux Etats-Unis, Linux 4U qui n'est pour l'instant distribuée qu'en Allemagne et Linux FT, que l'on trouve en Grande-Bretagne. La recette qui suit marche avec Unifix 1.7, Linux 4U 1st Edition, Linux Universe 2nd Edition et Linux FT 1.1.

- Copiez `linux-x86.jdk.static-motif.tar.gz` que l'on trouve à l'adresse : ftp://www.blackdown.org/pub/Java/linux/ dans le répertoire /usr/local, puis restaurez l'archive (grâce à gzip et tar) ;
- Modifiez la première ligne du script `.java_wrapper` `PRG=‘which $0‘ >/dev/null 2>&1/ ennbsp;:`

  `PRG=‘type $0 | cut -d ’ ’ -f 3‘ >/dev/null 2>&1/`;
- Récupérez `libc-5.2.18.bin.tar.gz` (cela fonctionne aussi avec `libc-5.2.16.bin.tar.gz`), puis restaurez cette archive dans le répertoire / ;
- Récupérez `ld-so.1.7.12.tar.gz` sur `tsx-11.mit.edu` (ou un site miroir) et restaurez-le dans /usr/local. Placez-vous dans le répertoire créé et lancez instldso.sh. Ceci exécutera également ldconfig. Il est probable que des versions plus récentes de ld-so fonctionneront également ;
- Ajoutez /usr/local/java/bin à la variable d'environnement PATH ;
- Récupérez netscape-v20b4-export.i486-unknown-linux.tar.Z, et restaurez-le dans /usr/local/<votre_Repertoire_Netscape
- Créez un lien de /usr/local/bin vers ../<votre_Repertoire_Netscape>/netscape ;
- Placez /usr/local/<votre_Repertoire_Netscape>/moz2_0.zip dans /usr/local/lib/netscape/.

Une alternative à la modification du PATH est, selon Ralf Strobel, de :

- Copier les scripts .java_wrapper et appletviewer, ainsi que les liens @java, @javac, @javadoc, @javap et @jdb de /usr/local/java/bin vers /usr/local/bin, de façon à pouvoir les utiliser depuis n'importe où ;
- Dans /java/bin/i586 vous pouvez ajouter un script appelé 'javadoc'

  #!/bin/sh ‘dirname $0‘/java sun.tools.javadoc.Main -d <votre_Repertoire_HTML> $

  où, si vous êtes l'utilisateur jrandom, <votre_Repertoire_HTML> correspond à  jrandom/.java-html.

# 6 Visualisation des applets

Pour lancer une démonstration sans passer par l'appletviewer, vous pouvez lancer la commande suivante depuis le haut de l'arborescence Java (/usr/local/java sur la plupart des machines) :

```
bin/java sun.applet.AppletViewer demo/<nom_repertoire>/exampleN.html
```

où <nom_repertoire> est le répertoire de la démonstration et N le numéro du fichier d'exemple (certains répertoires en contiennent plusieurs).

Steve Green raconte qu'il a testé les tutoriels disponibles sur le site java de SUN et sur celui du NTMUG. Son expérience révèle quelques problèmes dans la syntaxe du tutoriel de Sun. C'est pourquoi il est passé à celui du NTMUG.

L'application appletviewer requiert comme argument un fichier HTML contenant un marqueur APPLET.

Par exemple, si votre fichier HTML Hello.html ressemble à :

```
<HTML>
<HEAD>
<TITLE> Programme Test Hello {$<$}/TITLE>
{$<$}/HEAD>
<BODY>
Ceci est un test de l'appletviewer
<APPLET CODE="Hello1.class" WIDTH=150 HEIGHT=25>
{$<$}/APPLET>
{$<$}/BODY>
{$<$}/HTML>
```

La commande `appletviewer Hello.html` vous montrera l'applet. L'avantage de l'appletviewer est que les événements envoyés à l'applet (start(), init(), etc) sont conformes aux spécifications de Sun, ce qui n'est pas le cas de Netscape 2.0b4 (probablement à cause d'un bug).

L'un des inconvénients d'appletviewer est sa grande lenteur par rapport à Netscape.

Lorsque vous lancez Java pour la première fois, une fenêtre de licence similaire à celle de Netscape est affichée.

# 7 Problèmes connus

## 7.1 GNU make ne fonctionne plus après l'installation de Java

Les versions récentes de libc corrigent un bogue qui en cachait un autre dans la version GNU de make (le symptôme en est que make ne cherche plus les règles dans les Makefile). Un patch corrigeant ce problème est inclus dans la distribution 5.2.8 de libc.

## 7.2 Vous avez des erreurs `dirname: too many arguments`

Votre variable d'environnement CLASSPATH n'est pas initialisée correctement. Dans le fichier .java_wrapper, on trouve quelques lignes ressemblant à :

```
PRG=`which $0`
J_HOME=`dirname $PRG`/..
```

Malheureusement, la commande which généralement fournie avec Linux est horriblement défectueuse et certains shells risquent d'affecter à $0 le chemin complet. Selon Randy Chapman, la solution est d'utiliser soit :

```
J_HOME=`dirname $0`/..
```

soit, de façon plus sûre :

```
J_HOME=/usr/local/java
```

Un autre correctif, proposé par Dave Dittrich, est :

```
PRG=`csh -c "which $0"`
```

Quant à Tim Farnum, il propose de changer la ligne PRG=‘which $0‘ en :

```
PRG=$0
```

Lutz Behnke, enfin, suggère :

```
PRG=`type -path $0` >/dev/null 2>&1
```

Une modification similaire doit être appliquée au script d'appletviewer.

## 7.3 Vous avez des erreurs du type `cannot find class java/lang/Thread`

Votre variable d'environnement CLASSPATH n'est pas initialisée correctement. Voir plus haut.

## 7.4 Un message d'erreur fait référence à /dev/zero

Passez root et faites un `chmod 666 /dev/zero`.

### 7.5 SEGFAULT

Il peut arriver que votre écran se remplisse de messages d'erreurs, que le système remplisse joyeusement votre mémoire virtuelle, puis se bloque.

Dans ce cas, il vous manque probablement une bibliothèque quelque part. Lancez ldconfig -v et regardez ce qu'il manque. Il est également possible que les variables d'environnement LD_LIBRARY_PATH ou CLASSPATH ne soient pas positionnées. Enfin, il se trouve que certaines applets sont boguées et bloquent le JDK Linux (il y a d'ailleurs un moyen d'éviter le blocage en ayant un autre xterm dans lequel tourne top. Il suffit alors d'utiliser top pour tuer le processus java AVANT qu'il remplisse la mémoire virtuelle et bloque le système).

Java semble nécessiter d'importantes ressources. Il est donc conseillé de restreindre au maximum le nombre d'applications ouvertes ou qui s'exécutent. Il parvient à se charger sur un 486DX-2-75 avec 8 Mo de RAM et 16 Mo de swap (cela prend tout de même une minute). L'auteur a ainsi réussi à faire tourner deux applets animées simultanément (ou à peu près) avant que le système se retrouve à court de mémoire virtuelle et se bloque.

### 7.6 bin/java, bin/javac, ou bin/appletviewer vous renvoie un message d'aide

Vous avez oublié des paramètres de la ligne de commande.

### 7.7 Les applets apparaissent bien dans l'appletviewer, mais pas lorsqu'elles sont placées sur un serveur Web.

Une erreur classique ayant cette conséquence est d'associer aux applets un type MIME incorrect. Votre serveur doit renvoyer avec l'applet un en-tête indiquant qu'elle est du type MIME `text/plain`, `application/octet-stream` ou tout autre type pour lequel le client ne dispose pas d'un module de traitement spécial. La manière de remédier à cela dépend du serveur que vous utilisez. (John Franks)

Il semble également que `tinyhttpd`, un serveur HTTP écrit en Perl, renvoie un `content type` incorrect. Apache, en revanche, semble fonctionner relativement correctement.

### 7.8 Fichier de traces

Selon Joey Oravec, HotJava conserve un fichier de traces des actions qu'il accomplit et des éventuels problèmes qu'il rencontre. Si vous devez établir un diagnostic par vous-même, examinez le fichier $HOME/.hotjava/weblog, dans le répertoire racine de l'utilisateur. Ce fichier vous permettra de repérer plus aisément s'il vous manque une bibliothèque ou s'il s'agit d'un problème similaire.

## 8 Ressources associées

Les informations générales concernant Java peuvent être trouvées dans la FAQ publiée dans le newsgroup comp.lang.java et disponible à l'adresse http://www.city-net.com/ krom/java-faq.html.

Un autre document du type FAQ, plus ancien, a été écrit par Joey Oravec joey@sun.science.wayne.edu. On peut le trouver à l'adresse http://www.science.wayne.edu/ joey/java/linux.html. Il traite principalement de la version alpha, mais il se pourrait qu'il fusionne avec la FAQ citée ci-dessus.

Voici une liste de quelques pages WWW traitant de Java :

- Karl Asha's General Linux Java Page ;
- LinuxJava Mailing List ;
- Netrek for Java ;
- Blue-Skies for Java ;
- Java Hints Page.

# 9 A ajouter...

- Plus de recettes propres aux différentes déclinaisons de Linux.

# 10 Remerciements

L'auteur adresse ses plus vifs remerciements à tous ceux qui ont contribué à ce document, en particulier à :

- John Franks <john@math.nwu.edu> ;
- Zachary DeAquila <zachary@zachs.place.org> ;
- Steve Greene <sgreene@access.digex.net> ;
- Dave Dittrich <dittrich@cac.washington.edu> ;
- Dave Flanagan <dave@ora.com> ;
- Joey Oravec <joey@sun.science.wayne.edu> ;
- Adam Smith <aws@cs.brown.edu> ;
- Joe Buck <jbuck@Synopsys.com> ;
- Omar Loggiodice <ologgio@netdepot.com> ;
- Stefan Middendorf <mdorf@stud.fh-heilbronn.de>.

La présente version de ce document inclut le mini-HOWTO de Steve Greene pour Red Hat/Caldera, souvent annoncé mais jamais publié. L'auteur a également pêché quelques informations utiles sur la page WWW de Joey Oravec – grâces lui en soit rendues.

La liste des autres HOWTOs et FAQs maintenus par l'auteur peut être consultée sur sa page d'accueil http://www.locke.ccil.org/ esr/home.html.

La liste des HOWTOs en français peut être consultée sur http://www.loria.fr/linux/.

Le traducteur, enfin, souhaite remercier Pierre Vassellerie (Pierre.Vassellerie@obspm.fr) et Bernard Choppy (choppy@imaginet.fr) pour leur aide dans la mise en forme et la correctionde ce document.