

# Sun rend Java disponible sous licence open source

par [La rédaction Java](#)

Date de publication : 13/11/2006

Dernière mise à jour : 13/11/2006

Sun vient de rendre Java Open Source. Quelle est la licence qui a finalement été choisie ? Quelles en seront les conséquences pour les utilisateurs de la JRE ? Et pour les développeurs ? Vous trouverez toutes les réponses à vos questions dans cet article

I - Remontons le temps

- I-A - JavaOne, Mai 2006
- I-B - LinuxWorld, Août 2006
- I-C - Ces derniers jours
- I-D - Aujourd'hui

II - Le communiqué de presse

- II-A - Le titre du communiqué de presse
- II-B - Sous-titre du communiqué de presse
- II-C - 1er paragraphe
- II-D - 3ième paragraphe
- II-E - 4ième paragraphe
- II-F - Java SE
- II-G - Java ME
- II-H - Java EE
- II-I - NetBeans et les outils de développement Sun

III - Ce que le communiqué de presse ne dit pas

- III-A - Présentation de Tom Marble
- III-B - Le choix de la licence

Sous quelle licence sera donc disponible OpenJDK ?

Mais la licence GPL est "virale". Je vais donc devoir rendre mes programmes Java GPL ?

Tout le monde ne sera pas content. Les votes sur java.net étaient en faveur d'une licence Apache. Qu'allez vous leur répondre ?

III-C - Les sources

Nulle part dans le communiqué de presse il n'est précisé quelle est la version du JDK qui sera sous licence GPLv2. Pourriez-vous nous donner des détails ?

Est-ce à dire qu'il n'y aura plus de nouvelle mise à jour du JDK 5 ? Car si le JDK 6 est sous licence GPLv2 et que je contribue au code, il ne pourra jamais être "backporté" vers le JDK 5.0.

Comment est-ce que je pourrai accéder aux sources ?

Comment pourrais-je compiler les sources du JDK ?

Et plus tard, comment je ferai ? Car actuellement, il n'existe pas de plugin Mercurial pour NetBeans

Maintenant que le JDK est rendu open source, est-ce que je pourrai publier librement des Benchmarks de performances ? (car jusqu'à maintenant, il y avait certaines restrictions, comme obtenir au préalable l'accord de Sun)

III-D - Le report des bogues

Quel est l'outil qui a été choisi pour reporter les problèmes ?

III-E - Le JCP

Le JCP continuera-t-il à jouer un rôle ?

Faut-il apporter des modifications au JCP pour gérer le nouveau projet OpenJDK ? Si oui, lesquelles ?

III-F - Se protéger du Fork

Comment Sun va-t-elle protéger sa marque Java, malgré la mise en licence GPL du JDK ?

III-F-2 - Le TCK est il également disponible sous licence Open Source ?

III-F-3 - Sera-t-il disponible gratuitement ?

III-F-4 - Mais comment malgré tout éviter le fork ?

III-G - Community Leader

III-G-1 - Quel sera le rôle de Roger Brinkley dans cette affaire ?

III-H - Les fuites

III-H-1 - Savez-vous d'où sont venues les fuites ?

III-I - Remerciements

IV - Résumé

V - Conclusion

Cet article est également disponible au [format PDF](#) ([miroir http](#) )

## I - Remontons le temps

Le fait que Sun annonce qu'il va rendre Java Open Source n'est pas récent. La rumeur existe depuis des années. La question revenait à chaque JavaOne Et il n'y avait pas un interview de James Gosling où la question était posée.

Et puis, cette année, (ou l'année passée, je ne sais plus, le temps passe si vite), Jonathan Schwartz ([son blog, en français](#)) est devenu CEO de Sun. Et a apparemment réussi à convaincre Rich Green pour revenir travailler chez Sun.

### I-A - JavaOne, Mai 2006



Et voilà qu'on les retrouve tous les 2 à JavaOne. (Vous pouvez retrouver une vidéo de cette fameuse discussion sur [http://news.com.com/1606-2\\_3-6072875.html?tag=ne.vid](http://news.com.com/1606-2_3-6072875.html?tag=ne.vid)) Et Jonathan lui pose la question fatale : Allez-vous rendre Java Open Source ?

Gros sourire de Rich. Qui indique qu'il est très content d'être revenu. Et qui précise ensuite que la bonne question à poser maintenant, ce n'est plus SI on va le rendre open source, mais QUAND on va le rendre open source.

[Ce petit dialogue](#) a fait toute la une des médias consacrés à Java et même les autres.

Et puis, plus rien. Comme si tout cela était tombé dans l'oubli. Certains se demandaient finalement s'ils avaient été sérieux.

## I-B - LinuxWorld, Août 2006

Et puis voilà. [La rumeur se propage](#): **Sun va faire une annonce lors du salon LinuxWorld de San Francisco.**

LinuxWorld: Tout un symbole. Linux est sûrement l'un des plus célèbres exemples de réussite d'un logiciel open-source.

Et Sun a toujours désiré que Java se retrouve sur les distributions Linux.

Il y avait eu une tentative avec l'annonce de la DJL (Distribution Java Licence) à JavaOne, qui fut bien accueillie par certaines distributions, comme Ubuntu.

Mais d'autres, comme RedHat, continuaient de refuser de distribuer Java, vu que ce n'était toujours pas une licence open source.

Il était donc quelque peu normal que Sun fasse une déclaration quant à l'avancement de la mise en open source lors de ce salon.

Rich Green y a annoncé que les premières parties du JDK seraient disponibles pour la fin d'octobre 2006. Que ces parties seraient la technologie HotSpot et le compilateur Javac.

Il a également précisé lors de cet événement, que Java serait sous une licence open source reconnue par l'OSI.

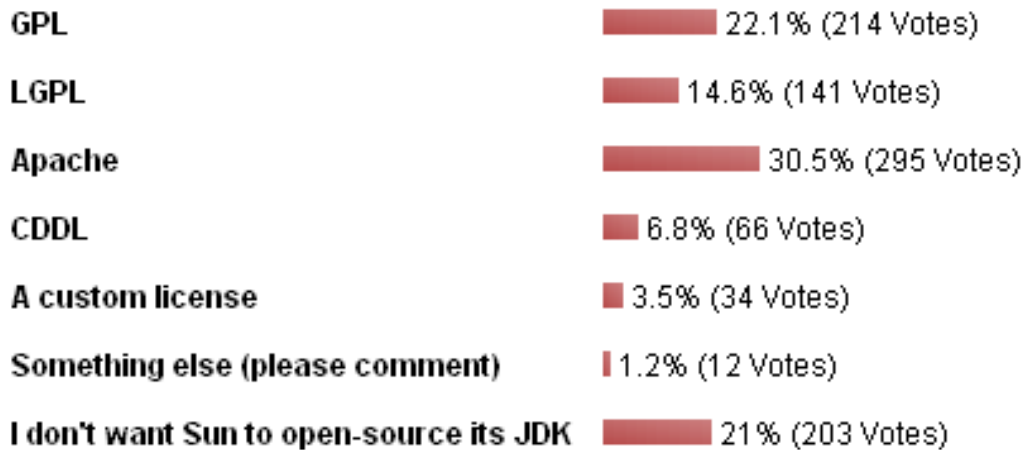
Et un nouveau projet a été créé pour l'occasion sur java.net.

(Retrouvez [la vidéo de cette interview sur YouTube](#))

A nouveau les rumeurs allaient bon train. Des sondages ont été lancés un peu partout, pour savoir le genre de licence que les gens aimeraient avoir, et pourquoi. Voici le résultat de 2 sondages :

# POLL RESULTS

What license would you like Sun's open-source JDK to use?



**Total Votes: 965**

*sondage sur java.net*

Affichage des résultats du sondage: Sous quelle licence Java devrait-il être disponible ?			
GPL		19	29,23%
LGPL		15	23,08%
Apache		9	13,85%
CDDL		16	24,62%
BSD		5	7,69%
Mozilla		0	0%
EPL		0	0%
MIT License		2	3,08%
Autre licence		1	1,54%
Pas de préférence		9	13,85%
Java ne devrait pas être Open Source		6	9,23%

Sondage à choix multiple Votants: 65. Vous avez déjà participé à ce sondage.

*sondage sur java.developpez.com*

On a également vu des réactions de certains acteurs du marché à ce sujet.

Mais un doute continuait à nous trotter dans la tête: Toutes les licences Open Source reconnues par l'OSI ne sont pas des licences libres reconnues par la Free Software Foundation.

## I-C - Ces derniers jours

Et puis, à nouveau le calme. La date de fin octobre approche. On apprend que ce sera finalement pour plus tard, mais sans vraiment plus de détails.

Et puis récemment, tout s'emballe, [une rumeur se répand aussi vite que l'éclair sur le Net](#): Sun aurait choisi une licence GPL. Les uns sont contents, les autres assommés.

Certains se demandent alors quelle sera la version de la licence GPL. La 2, ou la 3 ? Car la 3 pourrait être compatible avec la licence Apache. Et donc sauver quelque peu le projet Harmony.


Quasi simultanément, Roger Brinkley nous apprenait sur [son blog](#) qu'il arrêterait son rôle de community leader pour le JavaDesktop. Car il allait revêtir le rôle de community leader pour une toute nouvelle communauté qui devrait combiner les développeurs d'applications et de plateforme.

Allait-il devenir le community leader de la version open source de java ?

## I-D - Aujourd'hui

Voilà. Aujourd'hui, le grand jour est arrivé. Sun a mis fin aux rumeurs. Le communiqué de presse est enfin disponible.

On est maintenant fixé sur le sort que Sun a réservé à Java pour les prochaines années.

 ***L'avenir de Java se jouera très certainement sur le choix de la licence choisie par Sun, mais aussi sur la façon dont Sun va coopérer avec la communauté open source.***

Découvrons maintenant ce communiqué de presse.

## II - Le communiqué de presse

Tout d'abord, vous trouverez le communiqué de presse sur <http://www.sun.com/smi/Press/sunflash/index.xml>

Nous allons maintenant en analyser les points les plus importants.

### II-A - Le titre du communiqué de presse

#### Le titre du communiqué de presse



Sun Open Sources Java Platform and Releases Source Code Under GPL Licence Via NetBeans and Java.net Communities

La rumeur était donc bien fondée: Sun rend bien le code **Java sous licence GPL**.

Et le code Java sera disponible via les communautés [NetBeans](#) et [Java.net](#)



*Mais que vient faire NetBeans dans cette histoire ?*

### II-B - Sous-titre du communiqué de presse

#### Sous-titre du communiqué de presse



Builds on Success of NetBeans and OpenSolaris to Establish New Communities For Open Source Development of Java Technology

Sun nous fait savoir que si Java est maintenant disponible sous une licence GPL, c'est parce qu'ils ont pu s'appuyer sur leurs expériences tirées du succès de la mise en Open Source de NetBeans et d'OpenSolaris.

### II-C - 1er paragraphe

### 1er paragraphe



Sun Microsystems, the creator and leading advocate of Java(TM) technology, today announce it is releasing its implementations of Java technology as free software under the GNU General Public License version two (GPLv2). Available today, are the first pieces of source code for Sun's implementation of Java Platform Standard Edition (Java SE) and a buildable implementation of Java Platform Micro Edition (Java ME). Details are available at <http://www.sun.com/opensource/java>. In addition, Sun is adding the GPLv2 license to Java Platform Enterprise Edition (Java EE), which has been available for over a year under the Common Development and Distribution License (CDDL) through Project GlassFish(TM) at <http://glassfish.dev.java.net>.

On y apprend ici que le code Java sera donc disponible **sous licence GPLv2**.

***Cela met également fin aux débats GPLv2 / GPLv3. La GPLv3 n'est actuellement pas disponible. Il était hors de question d'attendre que la GPLv3 soit prête pour rendre Java Open Source.***

Pour ce qui est de Java SE, tout n'est actuellement pas disponible aujourd'hui.

***Cela se fera au fur et à mesure, mais tout Java SE devrait être sous licence GPLv2 pour JavaOne 2007 au plus tard.***

Egalement intéressant à noter est le fait que **Glassfish sera disponible sous une double licence GPLv2/CDDL**.

***Java EE est la seule des trois plateformes Java (SE, EE, et ME) à être disponible sous deux licences open source.***

### II-D - 3ième paragraphe

### 3ième paragraphe



"By open sourcing Sun's implementation of Java technology, we will inspire a new phase of developer collaboration and innovation using the NetBeans(TM) Integrated Development Environment (IDE) and expect the Java platform to be the foundation infrastructure for next generation Internet, desktop, mobile and enterprise applications," said Rich Green, executive vice president of Software at Sun. "With the Java Development Kit (JDK(TM)) released as free software under the GPL, Sun will be working closely with distributors of the GNU/Linux operating system, who will soon be able to include the JDK as part of the open source repositories that are commonly included with GNU/Linux distributions."

En gros, Sun veut projeter son EDI, NetBeans, en avant, et espère qu'on retrouvera Java partout. Aussi bien dans les applications mobiles que dans la prochaine génération d'internet, en passant par le PC de bureau et les serveurs.

Ils collaboreront également avec les distributions Linux pour que le JDK se retrouve bien dans toutes ces distributions.

## II-E - 4ième paragraphe

### 4ième paragraphe



"Everyone has been expecting that one day Sun would open source Java technology, but no one expected just how far they'd go -- GPL. A bold move, and a great opportunity both for Sun and for free and open source software, " said Tim O'Reilly, founder and CEO of O'Reilly Media.



On s'attendait à ce que cela se fasse un jour, mais personne ne s'attendait à ce qu'ils aillent si loin.



Une grande opportunité aussi bien pour Sun que pour le logiciel libre et open source.

## II-F - Java SE

### Java SE



Sun is releasing three significant software components today for the ongoing development of Sun's open source implementation of Java SE in the Java.net community: Java HotSpot(TM) technology, the Java programming language compiler (javac(TM)) and JavaHelp(TM) software. Sun expects to release a buildable JDK in the first quarter of 2007, following established free software community practices for licensing virtual machines and their associated libraries. Java HotSpot technology and javac are two of the most important elements of Java SE; Java HotSpot technology is the Sun implementation of the Java Virtual Machine (JVM(TM)) and the core component of

## Java SE

the Java Runtime Environment (JRE), which translates Java code to the specific operating system and chip architecture, allowing Java software to run everywhere and javac is the compiler that analyzes Java source code for correctness and generates proper bytecodes for execution. JavaHelp software is the documentation system to complement the JDK.

These first components of the OpenJDK(TM) project will allow developers to experiment with the compiler, try out new language features, learn how a world-class virtual machine is built, port the JVM to new hardware architectures and operating systems, fix bugs and contribute new features. Through the OpenJDK project, developers will be able to directly influence the future of the JDK implementation, participate with their peers in an open community and help take Java technology where it hasn't been before.

Aujourd'hui, seuls trois composants significatifs du JDK sont disponibles:


- la technologie Hotspot(TM)
- le compilateur javac(TM)
- JavaHelp(TM)

**Un JDK complet devrait être disponible pour le 1er trimestre 2007.**

 ***Ces premières pièces du JDK devraient ravir ceux qui patientaient pour pouvoir rajouter le support des closures dans le compilateur, à des fins de prototypes, et récolter le feedback des utilisateurs.***

***On peut également espérer voir apparaître les premiers bugfixes au niveau de la technologie HotSpot.***

Sun a également donné un nom à son Java Open Source: OpenJDK

 *Un nom très original, lorsqu'on le compare à OpenSolaris, OpenSparc :-)*

*Pour la petite histoire, ce nom figure depuis longtemps sur [le BOF qui se tiendra à Javapolis le mercredi soir](#)*

## II-G - Java ME

### Java ME



Available immediately in the Java.net community, is the source code for Sun's feature phone Java ME implementation, the next generation version of the platform that currently enables rich mobile data services in over 1.5 billion handsets. Also available is Sun's source code for the Java ME testing and compatibility kit framework, the foundation for Sun's Java ME compatibility tests. Later this year, Sun will release additional source code including its advanced operation system phone implementation and the framework for the Java Device Test Suite.

## Java ME

Sun is releasing these technologies as free software in order to accelerate the development and evolution of the platform, reduce fragmentation and drive down development costs throughout the Java ME ecosystem. In addition, this move will provide easy access to the latest versions of Java ME platform technologies and, for the first time, enable the whole Java ME community to follow the activities of and participate in the development of these technologies.

Egalement disponible via la communauté Java.net, le code source de Java ME. .

 **Cela couvre aussi bien CDC que CLDC**

A noter également que le code source du TCK Framework pour Java ME sera également disponible.

Le but est d'accélérer le développement et l'évolution de la plateforme, réduire la fragmentation et les coûts de développement, en ayant un écosystème Java ME.

## II-H - Java EE

### Java EE



Sun is also announcing that it is now releasing the source code for Project GlassFish (part of the GlassFish Community) under a dual open source license. In addition to CDDL, Project GlassFish will also be available under GPLv2 in the first quarter of 2007. By adding a second license, we simplify the process of combining and distributing GlassFish code with other GPL licensed communities. By offering all of Java under a common license, developers can now more easily distribute updated versions of Java SE, Java EE and Java ME together.

Glassfish, l'implémentation open source de Java EE 5 menée par Sun sera la seule implémentation des 3 plateformes Java (Java SE, ME et EE) à être disponible sous une double licence open source: GPLv2 et CDDL.

A noter que **Glassfish ne sera disponible sous licence GPLv2 que dans le 1er trimestre 2007.**

## II-I - NetBeans et les outils de développement Sun

### NetBeans and Sun Development Tools

### NetBeans and Sun Development Tools



The NetBeans IDE can dramatically simplify getting started with JDK development because the open source components have already been configured as NetBeans projects. Developers can download the source code, open it in the NetBeans IDE, and use the Build Project command to build it. For further information and a step-by-step tutorial go to: <http://nb-openjdk.netbeans.org>. In addition, an application developer project is available as part of the Mobile & Embedded community, with links to resources such as the NetBeans Mobility Pack, the Java ME authoring tool that delivers a whole new level of sophistication and ease for drag-and-drop screen design. Sun is also making available the world-record producing Sun(TM) Studio development environment for the development of the platform-specific native code in the Java HotSpot virtual machine.

The recently announced NetBeans 5.5 contains a variety of new features, including: Java Persistence API and JAX WS 2.0 productivity tools, Subversion support, and enhancements to the NetBeans GUI Builder (formerly known as Project Matisse). NetBeans 5.5 is the first and only freely available IDE to provide comprehensive support for Java EE 5, the industry standard for developing portable, robust, scalable, and secure server-side Java applications.

Nous savons donc maintenant pourquoi Sun mentionnait NetBeans dans le titre de sa Press Release. Tout a été fait pour que l'on puisse compiler le JDK depuis NetBeans.

En effet, il sera *facile* de pouvoir compiler tout le JDK depuis NetBeans, car le code source que l'on télécharge sera déjà sous la forme de projet NetBeans.

Tous les détails se trouvent sur <http://nb-openjdk.netbeans.org>.



**Une autre URL à rajouter dans vos favoris: <http://nb-openjdk.netbeans.org>**

### III - Ce que le communiqué de presse ne dit pas

Bien que le communiqué de presse nous révèle le type de licence (GPLv2), le nom du projet (OpenJDK), et les URL importantes, il y a des points sur lesquels il est muet: comme par exemple l'outil de versionning qui a été retenu, comment se fera le suivi des bugs, Quid du TCK, ..., ... .

Mais nous avons une excellente nouvelle pour vous:

#### **Nous avons pu obtenir une interview avec Tom Marble.**

Alexis Moussine-Pouchkine (Sun France), bien connu par ceux qui fréquentent les forums de [java.developpez.com](#), était également présent lors de cette interview.

#### **Cette interview concerne principalement le JDK de Java SE.**

### III-A - Présentation de Tom Marble

Tom Marble est francophone. Il travaille sur la mise en Open Source de Java avec Simon Phipps et son équipe.

C'est également lui qui a permis la mise en place de la DJL (Distribution Java Licence) qui permet à Ubuntu de redistribuer Java en version binaire.

Retrouvez Tom Marble sur [son blog](#).

Simon Phipps quant à lui doit gérer la stratégie open source de Sun, de OpenOffice à Java en passant par NetBeans, OpenSPARC, ou encore OpenSolaris.

Il s'occupe en particulier du choix des licences et des relations avec les autres communautés Open Source: Apache, FSF, Gnome, Debian, etc...

### III-B - Le choix de la licence

#### Sous quelle licence sera donc disponible OpenJDK ?

OpenJDK sera disponible sous licence GPLv2.

La licence GPLv3 a été écartée de suite, puisqu'elle n'est actuellement pas disponible. Et on ne peut se permettre d'attendre qu'elle le soit.

A noter que Sun continuera également à fournir Java sous licence commerciale à ses partenaires.

#### Mais la licence GPL est "virale". Je vais donc devoir rendre mes programmes Java GPL ?

Non, car nous y avons rajouté une exception concernant le classpath.

*Le projet GNU Classpath a également rajouté une pareille restriction. Voir <http://www.gnu.org/software/classpath/license.html>*

Cela permettra à tout le monde de continuer à choisir la licence qu'il désire pour son programme Java.

Tout le monde ne sera pas content. Les votes sur java.net étaient en faveur d'une licence Apache. Qu'allez vous leur répondre ?

Vous savez, ce genre de décision ne sont pas prise à la légère. Nous avons eu beaucoup de discussion avec les différents acteurs du marcher, y compris la fondation Apache et le projet Harmony.

### III-C - Les sources

Nulle part dans le communiqué de presse il n'est précisé quelle est la version du JDK qui sera sous licence GPLv2. Pourriez-vous nous donner des détails ?

Le code source qui vous est donné sous licence aujourd'hui concerne le code source du JDK 6. Mais tous nos efforts se porteront principalement sur le JDK 7.

Est-ce à dire qu'il n'y aura plus de nouvelle mise à jour du JDK 5 ? Car si le JDK 6 est sous licence GPLv2 et que je contribue au code, il ne pourra jamais être "backporté" vers le JDK 5.0.

En fait si. Car, tout comme cela se fait aujourd'hui pour le JDK actuel, ainsi que d'autres projets open source appartenant à Sun, toute personne qui voudra voir son code intégré dans le code source du JDK devra signer une Sun Contribution Agreement.

*La Sun Contribution Agreement est déjà utilisée pour d'autres projets comme NetBeans, OpenOffice, Glassfish, OpenSolaris, ...*

Vous trouverez la liste des signataires sur cette page: [http://www.sunsource.net/CA\\_signatories](http://www.sunsource.net/CA_signatories)

*Ce document est valable pour tous les projets open source appartenant à Sun. Vous ne devrez le signer qu'une seule fois.*

Par ce document, vous acceptez en fait que Sun possède également le copyright sur votre contribution. Et Sun peut dès lors également l'utiliser pour le mettre dans le JDK 5 qui restera lui sous une licence qui ne soit pas open source.

Il nous sera donc toujours possible, si jamais quelqu'un apporterait une contribution importante et qu'on juge qu'elle doit être "back-portée" vers la JDK 5.0, de le faire aussi librement que dans le passé.

Comment est-ce que je pourrai accéder aux sources ?

A partir d'aujourd'hui, les codes sources seront disponibles via Subversion. Mais en lecture seulement.

Nous offrirons des accès en écriture au code source plus tard, et via un autre outil de versionning: Mercurial.

*Mercurial est ce qu'on appelle un Distributed SCM, contrairement à CVS ou Subversion. Il se rapproche plus*

*de l'outil utilisé par Sun en interne pour le versionning du JDK.*

Vous trouverez plus d'information concernant Mercurial ici: <http://www.selenic.com/mercurial>

### Comment pourrais-je compiler les sources du JDK ?

Les sources du JDK seront disponibles sous la forme de projet NetBeans.

Il vous sera donc possible de compiler les sources à l'aide de NetBeans.

Mais, j'aime autant vous prévenir de suite, il vous faudra une machine très costaude pour pouvoir compiler les sources.

Aussi, l'équipe de NetBeans travaille en collaboration avec l'équipe du JDK pour avoir une meilleur modularisation du JDK dans le but d'en simplifier sa compilation.

Mais il vous faudra patienter un peu. Vous savez, le timing pour mettre tout cela en place était vraiment très serré.

### Et plus tard, comment je ferai ? Car actuellement, il n'existe pas de plugin Mercurial pour NetBeans

Rassurez-vous ! L'équipe de NetBeans travaille actuellement à l'élaboration d'un module pour Mercurial.

Mais je ne sais pas vous donner de date.

### Maintenant que le JDK est rendu open source, est-ce que je pourrai publier librement des Benchmarks de performances ? (car jusqu'à maintenant, il y avait certaines restrictions, comme obtenir au préalable l'accord de Sun)

Je ne sais pas vous répondre à cette question. Je vais me renseigner.

### III-D - Le report des bogues

#### Quel est l'outil qui a été choisi pour reporter les problèmes ?

Actuellement, rien n'a changé. Vous reportez les problèmes de la même façon qu'auparavant, et vous en ferez le suivi comme auparavant.

### III-E - Le JCP

#### Le JCP continuera-t-il à jouer un rôle ?

Oui, tout à fait. De ce point de vue là, rien ne change. Le processus reste inchangé. Toute personne qui désirera apporter des améliorations à Java devra obligatoirement passer par le JCP. Comme maintenant.

Les prochaines versions du JDK continueront à être "menées" par un JSR, tout comme le fut les versions précédentes.

Faut-il apporter des modifications au JCP pour gérer le nouveau projet OpenJDK ? Si oui, lesquelles ?

Le JCP est en actuellement sous sa version 2.6. Et il permet déjà des implémentations open source des JSR. C'est d'ailleurs grâce à ces modifications, que l'on a pu voir apparaître des projets comme celui d'Harmony.

Il existe actuellement un JSR ([le JSR 306](#)) qui va apporter encore quelques modifications au JCP.

*Cela fait partie du processus normal du JCP. La version 2.5 du JCP date de 2002, et la version actuelle (2.6) du JCP date de 2004.*

Mais ces modifications ne sont pas requises pour que l'on puisse rendre le JDK 6 open source. C'est juste pour améliorer encore le processus.

### III-F - Se protéger du Fork

*Le Fork est le processus consistant à créer un produit dérivant de l'original. Bien souvent un fork a lieu lorsque la société à l'origine du produit ne répond pas assez rapidement aux besoins de ses clients, ou lorsqu'il y a des compagnies qui se disputent des parts de marché.*

### Comment Sun va-t-elle protéger sa marque Java, malgré la mise en licence GPL du JDK ?

Sun tient beaucoup à sa marque, Java. Et donc, les règles de Sun à ce sujet sont très claires :

Pour qu'une implémentation puisse dire qu'elle est *Java Compatible*, elle devra obligatoirement avoir passé le TCK.

Si elle n'a pas passé le TCK, elle ne pourra pas utiliser la marque Java.

### III-F-2 - Le TCK est il également disponible sous licence Open Source ?

Non. Le TCK n'est pas disponible sous licence Open Source.

### III-F-3 - Sera-t-il disponible gratuitement ?

Non. Le TCK ne sera pas disponible gratuitement à tout un chacun. Celui qui désirera passer le TCK devra l'acheter. Mais Sun pourra dans certains cas (comme cela s'est déjà fait dans le passé avec Jonas, par exemple) donner une bourse à un projet, pour qu'il puisse passer le TCK.

*Pour rappel, cela ne concerne que le TCK de Java SE.*

*Pour ce qui est de JAVA ME, Sun va fournir en open source non pas le TCK lui-même, mais le TCK framework, qui sert de socle pour les tests de compatibilité de Java ME de Sun.*

### III-F-4 - Mais comment malgré tout éviter le fork ?

Cela peut paraître quelque peu paradoxal, mais de toutes les licences Open Source, nous croyons que c'est la licence GPL qui pourra protéger au mieux le fork de Java.

Car en ayant choisi la licence GPL, nous ne pouvons empêcher le risque qu'un fork se produise, mais nous obligeons ceux qui prennent cette voie à devoir passer le TCK s'ils veulent continuer à utiliser la marque Java. Et nous obligeons également ceux qui seraient prêts à ne pas utiliser la marque Java, à devoir tout faire publiquement, et toujours sous licence GPL. Nous croyons vraiment que cela devrait être suffisant pour éviter le Fork de Java.

Par contre nous encourageons tous ceux qui désirent faire une implémentation du JDK sur une plateforme que nous ne supportons pas à le faire. Ce n'est certainement pas cela que nous empêcherons de faire. Mais il faudra tout de même qu'il passe le TCK pour pouvoir dire qu'il est Java Compatible.

### III-G - Community Leader

#### III-G-1 - Quel sera le rôle de Roger Brinkley dans cette affaire ?

*Pour rappel, Roger Brinkley avait annoncé sur son blog qu'il ne serait plus le community leader de JavaDesktop, mais qu'il conduirait une autre communauté.*

Roger Brinkley ne sera pas le community leader de la partie Java SE de OpenJDK. Il s'occupera de la communauté qui tournera autour de Java ME.

Pour ce qui est de la communauté Java SE, c'est Ray Gans, qui gère actuellement le projet Peabody, et qui s'occupe déjà du [portail Open Source pour la communauté JDK](#) qui sera le community leader pour la communauté OpenJDK Java SE.

*Le but du projet Peabody est d'améliorer la collaboration entre l'équipe du JDK de Sun et la communauté de développeurs.*

*Vous pouvez retrouver Ray Gans sur son blog: [http://weblogs.java.net/blog/ray\\_gans/](http://weblogs.java.net/blog/ray_gans/)*

### III-H - Les fuites

#### III-H-1 - Savez-vous d'où sont venues les fuites ?

Je préfère ne rien dire à ce sujet. Mais cela nous a tout de même permis de voir quelles seraient les réactions des acteurs du marché.

Et la réaction d'Anthony Green nous a conforté dans notre choix. Car c'est exactement la réaction que nous espérons avec la mise sous licence GPL d'OpenJDK.

*Anthony Green a [annoncé sur son blog](#) que si c'était vrai que Sun rendrait Java sous licence GPL, ils la livreraient avec la distribution Fedora.*

### III-I - Remerciements

Merci à tous les deux, Tom et Alexis, d'avoir pris la peine de répondre à nos questions.



## IV - Résumé

- Java SE, Java ME et Java EE sont maintenant disponibles sous licences GPLv2
- Sun a rajouté une Classpath Exception, comme la licence le lui permet, permettant à quiconque de faire tourner ses programmes Java sous la licence de leur choix
- Ils continueront également d'être disponibles sous la licence commerciale actuelle
- Glassfish sera sous licence GPLv2 ET CDDL
- Sont aujourd'hui disponibles en open source:

- la technologie HOTSPOT,

- le compilateur Java,

- et l'utilitaire JavaHelp

- Sun espère fournir un JDK "buildable" pour le 1er trimestre 2007
- Le projet s'appelle OpenJDK
- Il faudra obligatoirement passer le TCK pour pouvoir utiliser le nom Java
- Le TCK n'est pas sous licence Open Source
- Le JCP reste en place. Et continuera d'être utilisé pour rajouter des fonctionnalités aux plateformes Java
- il faudra signer une Sun Contribution Agreement si on désire que le code soit intégré dans le JDK/JRE
- Le code source sera tout d'abord disponible en lecture via Subversion, plus tard via Mercurial
- Le code source est disponible sous la forme de projets NetBeans.
- 2 adresses à retenir:

<http://sun.com/opensource/java>

<http://nb-openjdk.netbeans.org>

## V - Conclusion

Que dire, à part merci.

- Merci à Sun d'avoir créé ce magnifique langage qu'est Java
- Merci à Sun de l'avoir finalement rendu Open Source, et qui plus est sous licence GPLv2
- Merci à Sun de nous avoir fait confiance pour la réalisation de cette interview