

Date	25-3-2003
Auteur	CK
rév	4 16-01-2004

JAVA FOR 4DUMMIES

Java for 4Dummies	1
Présentation de java	2
Historique	2
JVM.....	2
J2SDK versus J2RE	2
Différentes éditions.....	2
versions	3
Vocabulaire Java	3
Méthodes statiques.....	3
Package.....	3
Source et classe.....	3
Types de programme Java.....	3
Applets.....	4
évolution.....	4
Installation de java	4
Sur Windows.....	4
sur MacOSX	4
Contrôle de l'installation de Java sur Windows	5
Abondance de bien ne nuit pas ?	6
Variable d'environnement PATH	6
Contrôle de l'installation de Java sur macOSX.....	7
Parallèle 4d/Java.....	7
Bibliographie	8

Pour commencer, expliquons-nous, pour les non-anglophones, sur le titre de ce document. Le titre anglais de la collection à succès « Pour les nuls » est « *For dummies* ». Ce mauvais jeu de mot situe donc le contenu, chargé de donner une culture générale sur Java pour développeurs 4D totalement ignorants de la question.

L'objectif est double :

- vous permettre de briller lors des réunions de développeurs 4D ;
- suivre sans problème de vocabulaire les articles ou notes techniques concernant la connectivité entre 4D et les applications Java.

Les lecteurs familiers avec cet environnement peuvent se passer, sans remords, de la lecture de cette courte présentation.

PRÉSENTATION DE JAVA

HISTORIQUE

Commercialisée en 1995, la plate-forme Java est fondée sur deux concepts clés :

- le fonctionnement sécurisé en réseau ;
- la volonté d'amener le même programme à tourner sans modification sur toute machine pouvant exécuter du code : ordinateur, périphérique grand public, électronique embarquée,...

Pour cela, les programmes ne sont pas compilés en langage machine, mais en *bytecode*, pseudo-code objet, intermédiaire entre le source et le binaire exécutable. La conversion de ce pseudo-code en binaire n'est effectuée qu'au moment de l'exécution, par l'intermédiaire d'une machine virtuelle dépendante du système d'exploitation hôte.

Ce système permet à Java d'être disponible sur Unix, Linux, MacOS et Windows.

Java, langage objet, fut conçu dans une optique de simplification tirant les leçons des difficultés rencontrées en programmation C++. La notion de *garbage collector* (« ramasse-miette »), en particulier, vise à rendre la gestion mémoire moins complexe : le développeur n'a plus à supprimer explicitement les objets créés, Java s'en charge lui-même automatiquement lorsqu'il détecte que ces objets ne sont plus utilisés. En outre, Java n'autorise pas la notion de pointeur.

L'une des raisons du succès de Java vient de la promotion effectuée par son créateur SUN, mais également de son modèle économique. Le kit de développement est disponible gratuitement et le *runtime* d'exécution est librement redistribuable.

JVM

La machine virtuelle, permettant l'exécution du bytecode, s'appelle JVM : *Java Virtual Machine*.

Sa présence ou non sur un ordinateur et, le cas échéant sa version, dépendent du système d'exploitation. Cela constitue l'une des faiblesses de la plate-forme Java à laquelle SUN tente de remédier au fil des versions, (voir plus loin par exemple l'introduction du concept de « java plugin »).

Une seconde faiblesse provient de l'exécution interprétée des programmes Java, entraînant des performances inférieures aux programmes en langage natif. Là-aussi, SUN essaye d'apporter des solutions avec les compilateurs JIT (*Just-in-Time*), compilant en langage machine au moment de l'exécution, afin d'optimiser les boucles et appels ultérieurs.

OS	pré-installé	version
MacOS 8/9	Oui	Java1
MacOS X	Oui	Java2
Windows	Variable	Java2

Variations de la JVM suivant les plate formes

Il est préférable de télécharger la JVM depuis le site de SUN pour s'assurer de disposer de la dernière version. Deux packages de téléchargement s'y trouvent : SDK et JRE.

J2SDK VERSUS J2RE

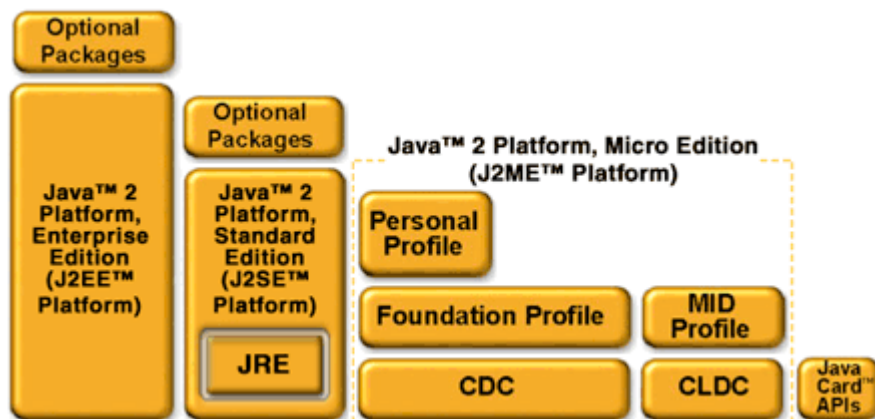
- JRE ou *Java Runtime Edition* : il s'agit du package d'utilisation de Java.
- JDK ou *Java Software Development Kit* : il s'agit du package destiné au développement d'applications Java, comprenant en particulier un compilateur.
Le SDK comprend le JRE.

DIFFÉRENTES ÉDITIONS

Trois éditions différentes sont proposées sur le site de SUN :

- J2SE : *Java 2 Platform, Standard Edition*, plate-forme pour bâtir des applications à destination des postes clients ;

- J2EE : *Java 2 Platform, Enterprise Edition*, plate-forme de développement à base de composants, les EJB (*Enterprise JavaBeans*) ;
- J2ME : *Java2 Platform, Micro Edition*, plate-forme de l'électronique grand public et embarquée.



La gamme Java, illustration provenant du site de SUN <file>java.overview.gif</file>

VERSIONS

La version courante au moment de la rédaction de cet ouvrage est la 1.4.x. Outre des performances améliorées de 20 à 80 % d'après SUN, elle offre un support intégré des standards XML comme SAX 2.0, DOM 2.0 et XSLT.

VOCABULAIRE JAVA

MÉTHODES STATIQUES

En Java, les méthodes peuvent être définies comme statiques. Une méthode statique appartient à la définition de la classe et ne nécessite pas ce que l'on appelle l'instanciation, c-à-d la création d'un objet particulier.

Des classes, uniquement composées de méthode statiques, servent ainsi de bibliothèques de routines, par exemple une classe d'utilitaires mathématiques.

PACKAGE

Il s'agit d'un regroupement de classes dans une archive au format zip. L'extension usuelle est « .jar » pour *Java Archive*. Un package se crée simplement grâce à un utilitaire de type « WinZip » ou par un appel en ligne de commande à « jar », un utilitaire Java.

SOURCE ET CLASSE

Un programme Java est composé de plusieurs fichiers sources au format texte, d'extension « .java ».

La compilation d'un fichier source crée un fichier d'extension « .class ».

TYPES DE PROGRAMME JAVA

Il existe trois types de programmes Java :

- **Applet** : chargé et exécuté au sein d'une page HTML ;
- **Stand-alone** : s'exécute comme une application autonome, tout en nécessitant la JVM ;
- **Servlet** : s'exécute côté serveur Web, équivaut aux scripts *server-side* d'autres technologies (ASP-PHP), le protocole HTTP des serveurs Web permet d'appeler les servlets.

APPLETS

L'aspect séduisant des applets réside dans leur téléchargement automatique depuis un serveur (qui n'est pas obligatoirement le serveur Web). Les applets sont ensuite exécutées en utilisant la JVM du navigateur. En principe, cela ne nécessite aucune installation sur le poste client.

En raison du retard des versions de JVM utilisées par les navigateurs, il vaut mieux concevoir une applet compatible avec la version 1 de Java.

Pour remédier à ce problème, SUN propose sur son site le **java plugin** : cette technique permet d'utiliser une JVM de SUN à jour, au lieu de celle obsolète du navigateur, qu'il s'agisse de Netscape Navigator ou de MS Internet Explorer. Si le plugin détecte que la JVM de SUN n'est pas installée, il propose automatiquement de la télécharger depuis le site de SUN : <http://java.sun.com/products/plugin/>.

La sécurité relative à l'exécution d'un programme téléchargé depuis un serveur constitue une préoccupation forte de SUN. Des mécanismes de sécurité de bas niveau limitent les risques, par défaut :

- les écritures sur le disque hôte sont très encadrées ;
- l'applet n'est pas autorisée à tenter une connexion vers une adresse IP différente de celle dont elle provient.

ÉVOLUTION

Les difficultés liées au retard des JVM des navigateurs, ajoutées à la lenteur de téléchargement des applets, ont rendu moins fréquent l'usage de ces dernières dans les pages HTML, au profit notamment des animations Flash reprenant le même modèle.

En revanche, les *servlets*, exécutées sur le serveur dans un contexte maîtrisé, sont totalement dégagées de ces deux freins. Leur utilisation connaît un réel succès.

Enfin, l'essor des micro-programmes, embarqués par exemple sur des *smart cards* (cartes intelligentes) ou des téléphones mobiles, insuffle un nouvel élan à la plateforme Java.

INSTALLATION DE JAVA

SUR WINDOWS

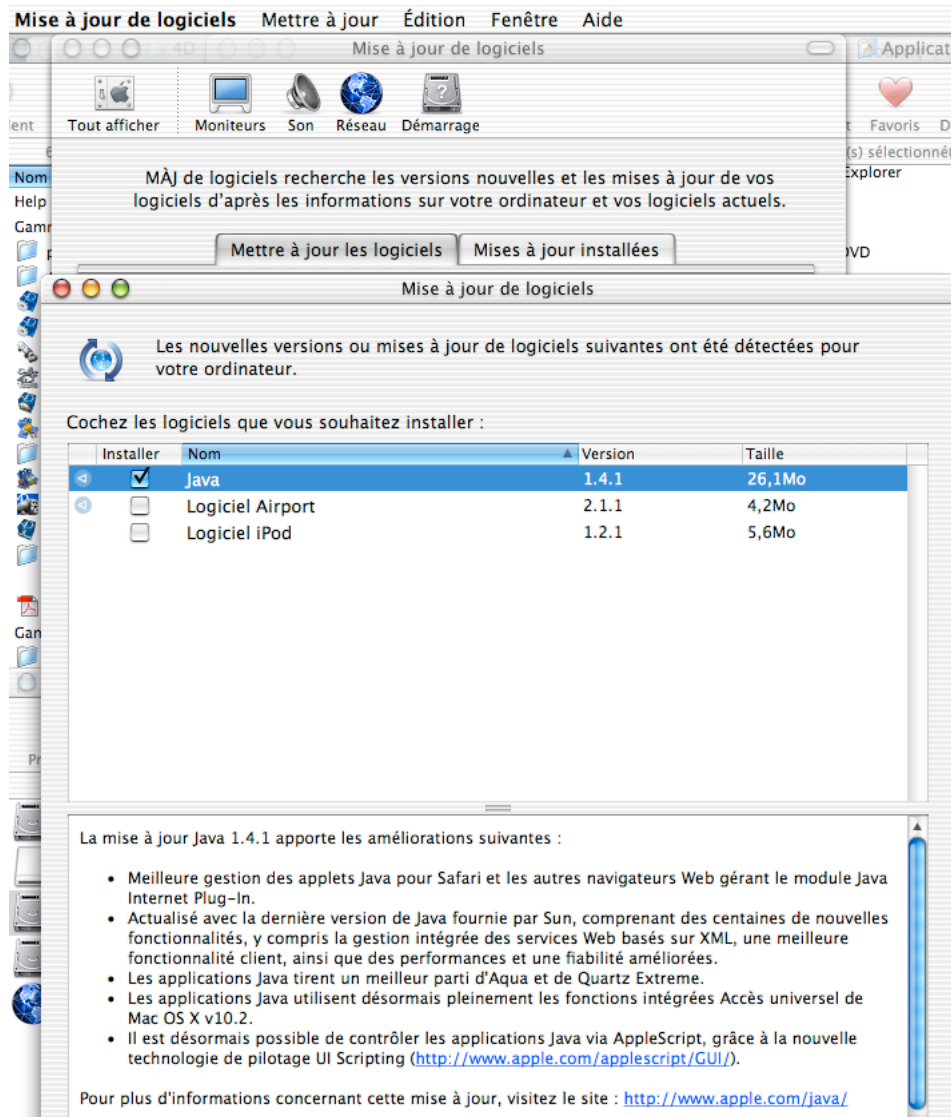
L'installation sur Windows dépend des versions de l'OS et de l'état du dernier procès entre Microsoft et SUN. Pour être certain de disposer de la dernière mouture, le téléchargement direct depuis le site de SUN s'impose : java.sun.com/getjava/.

SUR MACOSX

Java2 est installé par défaut, un bonheur.

Néanmoins, l'inscription gratuite au programme développeur d'Apple permet de suivre les mises à jour : www.apple.com/java/

Ce suivi est également proposé par la « Mise à jour de logiciels », partie intégrante du système d'Apple :

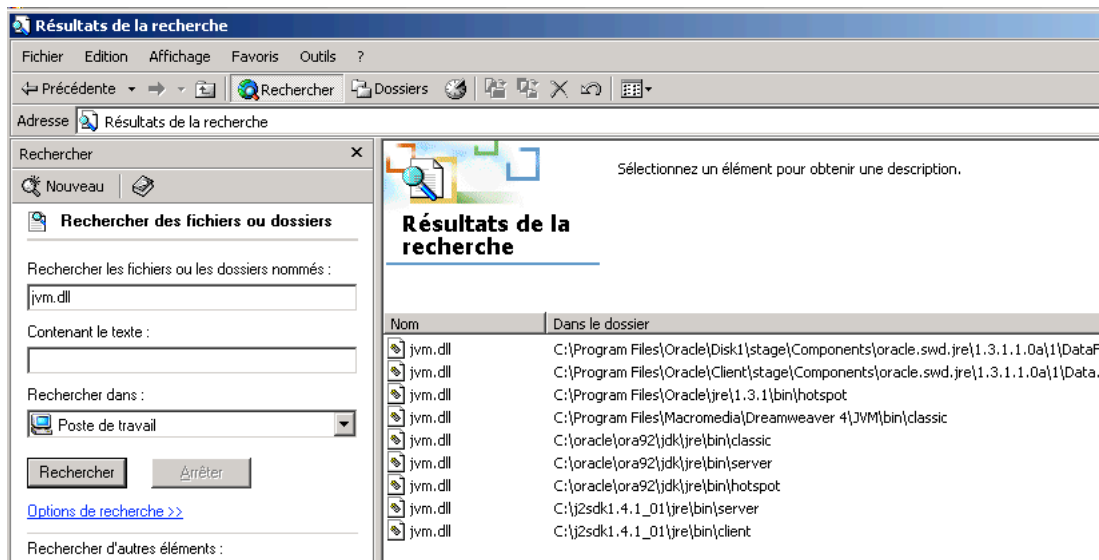


Mise à jour de Java sur MacOSX<file>majApplejava.pdf</file>

CONTRÔLE DE L'INSTALLATION DE JAVA SUR WINDOWS

Il est souvent important de connaître le chemin d'accès à la JVM. Suite aux bisbilles entre SUN et Microsoft, la JVM n'est plus systématiquement installée dans Windows. Chaque logiciel installe sa propre version dans son répertoire, difficile ensuite de s'y retrouver.

Voici par exemple le résultat d'une recherche portant sur « jvm.dll » sur un portable, où sont installés Oracle et DreamWeaver :



les machines virtuelles installées<file>JVM.tif</file>

ABONDANCE DE BIEN NE NUIT PAS ?

Encore faut-il connaître la version réellement utilisée par le système. Il suffit de taper `java -version` dans l'invite de commande.

```

Microsoft Windows [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>java -version
java version "1.3.1_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1_01)
Java HotSpot(TM) Client VM (build 1.3.1_01, mixed mode)

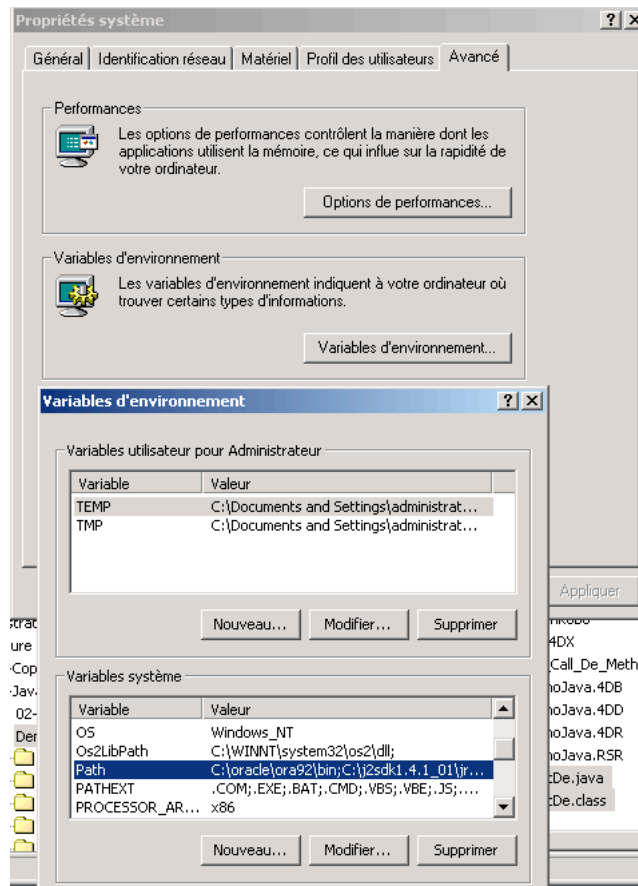
C:\>_
    
```

version de java utilisée<file>version.tif</file>

VARIABLE D'ENVIRONNEMENT PATH

Pour indiquer au système quelle JVM utiliser, son chemin d'accès doit figurer dans la variable d'environnement PATH.

Dans Windows 2000, l'accès à ce paramétrage s'effectue par un clic droit sur le poste de travail, puis par la sélection de « Propriétés » dans le menu contextuel. Le dialogue des propriétés système s'affiche, sélectionner alors l'onglet « Avancé ». Cliquer ensuite sur le bouton « Variables d'environnement... ». Le dialogue suivant apparaît :



Variables d'environnement<file>VarEnv.tif</file>

Sélectionner dans le groupe du bas, la variable Path, cliquer sur « Modifier... » et rajouter le chemin d'accès à la JVM désirée, après avoir éventuellement supprimé le chemin précédent. Pour paramétrer plusieurs chemins d'accès, il suffit de les séparer par des « ; ».

CONTRÔLE DE L'INSTALLATION DE JAVA SUR MACOSX

La saisie de `java -version` dans le terminal informe de la version courante sur la machine.



Contrôle de version de java sur OSX<file>JavaVersionOSX.tif</file>

PARALLÈLE 4D/JAVA

4D et Java se rejoignent par certains aspects :

- le choix dynamique du type d'interface graphique désiré (*look and feel* chez Java) est possible lors de l'exploitation;
- l'OS de la machine de développement ne conditionne pas le poste d'exploitation ;
- l'indépendance de plate-forme s'obtient par l'utilisation d'un *runtime* chez 4D en mode interprété et d'une *JVM* pour Java;

- en contexte d'applet, les modifications du programme sont automatiquement retransmises au poste client lors de la prochaine exécution, tout comme en mode client/serveur avec 4D.

Toutefois, des différences fondamentales existent :

- Java est avant tout un langage et n'intègre pas de base de données ;
- dans leur modèle économique respectif, le déploiement de solutions basées sur 4D est payant, celui de Java est gratuit;
- dans le mode de développement, 4D ne propose pas encore de langage-objet ;
- dans la rapidité et la souplesse de programmation qui reste nettement en faveur de 4D, même en utilisant des IDE (*Integrated Development Environment* : atelier de développement intégré) Java performants, du type de « JBuilder » de Borland.

BIBLIOGRAPHIE

Planète 4D

- #4 : « Ouverture sur Java » par Pascal Pradier, cet article propose une introduction de trois pages au produit 4D Open For Java, reprenant l'essentiel de la conférence donnée à l'Université 4D d'Arles en 2000.
- #7 : « Java voit 4D Write » par Christophe Keromen & Pascal Pradier, exemple 4D Open For Java permettant à une application Java de récupérer des documents 4D Write via une conversion RTF.

Note Technique 4D S.A.

- «4D s'ouvre sur Java » par Christophe Keromen, présente les concepts de base de 4D Open For Java avec un exemple de recherche et une description du *freeware* « *Java DataBase Viewer for 4D* ».

Site 4D S.A.

- « 4D in a Java World.pdf » : ce livre blanc de trois pages en anglais présente les différents principes d'intégration de 4D et Java.

Livres sur Java

- « Java in a nutshell » ed. O'Reilly : un ouvrage et une collection de référence.