

Installation GNAT OS X pour JVM

JVM-GNAT est l'environnement de développement Ada 2005 pour la plate-forme Java. À partir d'un code source en Ada, le compilateur génère un ByteCode compatible de l'environnement d'exécution Java 1.5 comme par exemple les applets.

1) Installation à partir des binaires pour Snow Leopard (GPL 2011)

La livraison GPL 2011 ne propose pas le compilateur prêt à l'emploi pour Mac. Nous pouvons soit le construire à partir des sources (voir paragraphe suivant) soit le prendre prêt à l'emploi sur Source Forge (ce que nous allons faire ici).

Télécharger le fichier suivant sur le bureau du Mac :
. Compilateur GNAT-JVM GPL 2011 : "gnat-jvm-gpl-2011-x86_64-apple-darwin10.8.0-bin.tgz", sur le site de Source Forge
"http://sourceforge.net/projects/gnuada/files/GNAT_GPL%20Mac%20OS%20X/2011-snow-leopard/".

Lancer le Terminal dans un compte administrateur et taper les commandes suivantes :

```
$ mkdir -p /usr/local/gnat-2011  
$ cd /usr/local/gnat-2011  
$ tar xzf ~/Desktop/gnat-jvm-gpl-2011-x86_64-apple-darwin10.8.0-bin.tgz
```

Le compilateur GNAT pour JVM s'installe à partir des répertoires /usr/local/gnat-2011/bin, lib et share.

Pour une utilisation courante, saisir aussi les commandes suivantes :

```
$ echo 'PATH=/usr/local/gnat-2011/bin:$PATH' >> ~/.profile  
$ echo 'PATH=/usr/local/gnat-2011/bin:$PATH' >> ~/.bashrc  
$ echo 'CLASSPATH=/usr/local/gnat-2011/lib/jgnat.jar:.' >> ~/.profile  
$ echo 'CLASSPATH=/usr/local/gnat-2011/lib/jgnat.jar:.' >> ~/.bashrc
```

Pour une utilisation temporaire, utiliser à chaque fois les commandes suivantes :

```
$ export PATH=/usr/local/gnat-2011/bin:$PATH  
$ export CLASSPATH=/usr/local/gnat-2011/lib/jgnat.jar:.
```

Une documentation aux formats HTML et PDF est disponible dans le répertoire `/usr/local/gnat-2011-src/share/doc` :

```
$ open /usr/local/gnat-2011/share/doc/html/jgnat_ug.html
$ open /usr/local/gnat-2011/share/doc/pdf/jgnat_ug.pdf
```

Un exemple de programme est disponible dans le répertoire `/usr/local/gnat-2011/share/examples/jvm`.

2) Construction du compilateur GNAT-JVM à partir des sources (GPL 2011)

La livraison 2011 ne propose pas le compilateur prêt à l'emploi pour Mac. Nous allons donc le construire à partir des sources.

Télécharger les fichiers suivants sur le bureau du Mac :
Compilateur 2011, jvm-windows : "gnat-gpl-2011-src.tgz" et "gnatvm-gpl-2011-src.tgz" sur le site "<https://libre.adacore.com>" à la page "Download GNAT GPL", ainsi que les correctifs apportés par Stephe Leake (<http://www.stephe-leake.org/>) complétés sur Blady :
<http://blady.pagesperso-orange.fr/telechargements/jgnat/jvm-gnat-gpl-2011.diff>

Lancer le Terminal et taper les commandes suivantes :

```
$ cd /usr/local
$ tar xzf ~/Desktop/gnatvm-gpl-2011-src.tgz
$ cd gnatvm-2011-src
$ tar xzf ~/Desktop/gnat-gpl-2011-src.tgz
$ mv gnat-gpl-2011-src/src/ada gnat_src
$ rm -fr gnat-gpl-2011-src
$ (cd gnat_src && patch < ../cil/frontend.difs)
$ patch < ~/Desktop/jvm-gnat-gpl-2011.difs
$ make -C jvm
$ make PREFIX=../jvm UG=jgnat_ug -C docs install
```

Le compilateur s'installe dans les répertoires `/usr/local/gnatvm-2011-src/jvm/bin`, `lib` et `share`.

Pour une utilisation courante, saisir aussi les commandes suivantes :

```
$ echo 'PATH=/usr/local/gnatvm-2011-src/jvm/bin:$PATH' >> ~/.profile
$ echo 'PATH=/usr/local/gnatvm-2011-src/jvm/bin:$PATH' >> ~/.bashrc
$ echo 'CLASSPATH=/usr/local/gnatvm-2011-src/jvm/lib/jgnat.jar:.' >> ~/.profile
$ echo 'CLASSPATH=/usr/local/gnatvm-2011-src/jvm/lib/jgnat.jar:.' >> ~/.bashrc
```

Pour une utilisation temporaire, utiliser à chaque fois les commandes suivantes :

```
$ export PATH=/usr/local/gnatvm-2011-src/jvm/bin:$PATH
$ export CLASSPATH=/usr/local/gnatvm-2011-src/jvm/lib/jgnat.jar:.
```

Une documentation aux formats HTML et PDF est disponible dans le répertoire `/usr/local/gnatvm-2011-src/share` :

```
$ open /usr/local/gnatvm-2011-src/jvm/share/doc/html/jgnat_ug.html
$ open /usr/local/gnatvm-2011-src/jvm/share/doc/pdf/jgnat_ug.pdf
```

Un exemple de programme est disponible dans le répertoire `/usr/local/gnatvm-2011-src/jvm/share/examples/jvm`.

3) Utilisation avec le Terminal

La commande "jvm-gnatmake" seule, sans paramètre, donne justement la liste des paramètres possibles. Néanmoins, la simple commande suivante donnera de bons résultats :

```
$ jvm-gnatmake hello
```

Le fichier `hello.adb` étant :

```
with Ada.Text_IO; use Ada.Text_IO;
procedure Hello is
begin
  Put_Line ("Hello from JVM World!");
end Hello;
```

Et les résultats ne se font pas attendre :

```
$ jvm-gnatmake hello
jvm-gnatcompile -c hello.adb
jvm-gnatbind -x hello.ali
jvm-gnatlink hello.ali
$ java hello
Hello from JVM World!
```

4) les commandes utiles avec le Terminal

La liste des commandes est obtenue de la façon suivante :

```
$ jvm-gnat
GNAT GPL 2011 (20110428)
Copyright 1996-2011, Free Software Foundation, Inc.
List of available commands
GNAT BIND      jvm-gnatbind
                réalise l'édition des liens des unités Ada compilées
GNAT CHOP      jvm-gnat Chop
                découpe un fichier en unités pour satisfaire les conventions Gnat
GNAT CLEAN     jvm-gnatclean
                nettoie les fichiers générés par gnat (non présent avec gnat-osx)
GNAT COMPILE   jvm-gnatcompile ou jvm-gcc
                compile une entité Ada
GNAT CHECK     jvm-gnatcheck
                vérifie le code source suivant des règles définies (non présent
                avec jvm-gnat-osx)

GNAT ELIM      jvm-gnatelim
                détecte et élimine les sous-programmes inutilisés (non présent avec jvm-
                gnat-osx)
GNAT FIND      jvm-gnatfind
                liste toutes les utilisations d'une entité Ada
GNAT KRUNCH    jvm-gnatkr
                réduit les noms de fichiers au nombre maximal de lettres spécifié
GNAT LINK      jvm-gnatlink
                réalise l'édition des liens de l'exécutable
GNAT LIST      jvm-gnatls
                liste le contenu des objets générés
GNAT MAKE      jvm-gnatmake
                utilitaire optimisé de compilation multi-unités
GNAT METRIC    jvm-gnatmetric
                statistiques sur le code Ada (non présent avec jvm-gnat-osx)
GNAT NAME      jvm-gnatname
                réalise la correspondance entre les unités Ada et les noms des fichiers
                lorsque ceux-ci ne sont pas au standard gnat
GNAT PREPROCESS jvm-gnatprep
                pré-processeur externe
GNAT PRETTY    jvm-gnatpp
                reformate le source Ada (non présent avec jvm-gnat-osx)
GNAT STACK     jvm-gnatstack
                calcul la taille de pile mémoire maximale théorique (non présent avec jvm-
                gnat-osx)
```

GNAT STUB **jvm-gnatstub**

 créé le squelette d'un corps d'une spécification (non présent avec jvm-gnat-osx)

GNAT XREF **jvm-gnatxref**

 utilitaire d'édition des références croisées

De même chacune des commandes exécutée sans argument affichera justement la liste des arguments possibles.

\$ jvm-gnatmake (extrait)

Usage: jvm-gnatmake opts name {[-cargs opts] [-bargs opts] [-larges opts] [-margs opts]}

 name is a file name from which you can omit the .adb or .ads suffix

gnatmake switches:

-a Consider all files, even readonly ali files

-b Bind only

-B Build, bind and link full project

-c Compile only

...

-D dir Specify dir as the object directory

...

-f Force recompilations of non predefined units

-F Full project path name in brief error messages

-i In place. Replace existing ali file, or put it with source

-jnum Use nnn processes to compile

-k Keep going after compilation errors

-l Link only

-m Minimal recompilation

-M List object file dependences for Makefile

-n Check objects up to date, output next file to compile if not

-o name Choose an alternate executable name

-p Create missing obj, lib and exec dirs

-Pproj Use GNAT Project File proj

-q Be quiet/terse

-R Do not use a run_path_option when linking

-s Recompile if compiler switches have changed

-u Unique compilation, only compile the given files

-U Unique compilation for all sources of all projects

-v Display reasons for all (re)compilations

-vPx Specify verbosity when parsing GNAT Project Files

-we Treat all warnings as errors

-wn Normal warning mode (cancels -we/-ws)

-ws Suppress all warnings

-x Allow compilation of needed units external to the projects

-Xnm=val Specify an external reference for GNAT Project Files

-z No main subprogram (zero main)

--GCC=command Use this compiler command
--GNATBIND=command Use this gnatbind command
--GNATLINK=command Use this gnatlink command

Project, Source and Library search path switches:

-aPdir Add directory dir to project search path
-aLdir Skip missing library sources if ali in dir
-Adir like -aLdir -aldir
-aOdir Specify library/object files search path
-aldir Specify source files search path
-ldir Like -aldir -aOdir
-I- Don't look for sources & library files in the default directory
-Ldir Look for program libraries also in dir
-nostdinc Don't look for sources in the system default directory
-nostdlib Don't look for library files in the system default directory
--RTS=dir specify the default source and object search path
--subdirs=dir real obj/lib/exec dirs are subdirs

To pass an arbitrary switch to the Compiler, Binder or Linker:

-cargs opts opts are passed to the compiler
-bargs opts opts are passed to the binder
-largs opts opts are passed to the linker
-margs opts opts are passed to gnatmake

Compiler switches (passed to the compiler by gnatmake):

-fstack-check Generate stack checking code
-fno-inline Inhibit all inlining (makes executable smaller)
-fpreserve-control-flow Preserve control flow for coverage analysis
-g Generate debugging information
-ldir Specify source files search path
-I- Do not look for sources in current directory
-O[0123] Control the optimization level

5) Les utilitaires spécifiques JVM

(Les utilitaires fonctionnent même avec des fichiers issus de sources Java ;-)

a) jvmlist

Affiche le contenu d'un fichier binaire .class ou une bibliothèque de classes .jar avec le nom des classes de leurs attributs, champs et méthodes. L'option -g fait afficher le ByteCode de chaque méthode Ada.

Exemple :

```
$ jvmlist hello.class
```

b) jvmstrip

Elimine toutes les informations de mise au point contenues dans un fichier binaire .class ou une bibliothèque de classes .jar.

Exemple :

```
$ jvmstrip hello.class
```

c) jarmake

Groupe les classes dépendante d'un fichier binaire .class ou une bibliothèque de classes .jar ou tout autre fichier ainsi que la bibliothèque jgnat.jar dans une seule bibliothèque. Cela permet de construire des programmes autonomes comme des applets.

Exemple :

```
jarmake -o tictactoe.jar tictactoe\$.typ.class
```

d) jvm2ada

Génère les spécifications Ada correspondant à un fichier binaire .class ou une bibliothèque de classes .jar.

Exemple :

```
$ jvm2ada hello.class
```

6) La bibliothèque JRE en Ada

La bibliothèque du JRE est utilisable avec des spécifications correspondantes aux classes Java générées par l'utilitaire jvm2ada.

Mais auparavant il faut décompresser les bibliothèques JRE de Mac OS X.

```
$ cd /usr/local/gnatvm-2011-src/jvm/lib
```

```
$ mkdir jre
```

```
$ cd jre
```

```
$ jar -xf /System/Library/Frameworks/JavaVM.framework/Classes/classes.jar
```

```
$ jar -Ocf uncompressed_classes.jar com java javax org sun sunw META-INF
```

```
$ rm -fr META-INF/ com/ javax/ sun/ java/ org/ sunw/
```

```
$ jar -xf /System/Library/Frameworks/JavaVM.framework/Classes/jce.jar
```

```
$ jar -Ocf uncompressed_jce.jar javax sun META-INF
```

```
$ rm -fr META-INF/ javax/ sun/
```

```
$ jar -xf /System/Library/Frameworks/JavaVM.framework/Classes/charsets.jar
```

```
$ jar -Ocf uncompressed_charsets.jar sun META-INF
```

```
$ rm -fr META-INF/ sun/
```

```
$ jar -xf /System/Library/Frameworks/JavaVM.framework/Classes/jsse.jar
```

```
$ jar -Ocf uncompressed_jsse.jar com javax sun META-INF
```

```
$ rm -fr META-INF/ com/ javax/ sun/
```

```
$ jvm2ada -jgnat "uncompressed_classes.jar" "-Luncompressed_jce.jar" "-Luncompressed_charsets.jar" "-Luncompressed_jsse.jar"
```

Certaines spécifications contiennent des conflits qu'il faut résoudre manuellement. Télécharger le correctif du site Blady à sauvegarder sur le bureau :

<http://blady.pagesperso-orange.fr/telechargements/jgnat/jre-10.6.diff>

```
$ patch < ~/Desktop/jre-10.6.diff
$ jvm-gnatmake -g *.ads
# c'est long ... très long ...
```

La bibliothèque s'utilise comme toute bibliothèque GNAT :

```
jvm-gnatmake -I/usr/local/gnat-2011/lib/jre tictatoc
```

Le mois prochain : Les applets.

Pascal Pignard, juin-juillet 2009, novembre 2010, décembre 2011.