

Installation GNAT OS X 10.2

1) Installation du compilateur (version gcc-3.1-1173)

Pour pouvoir utiliser GNAT, il est indispensable d'avoir installé le compilateur version gcc-3.1-1173 avec les derniers outils de développement "December 2002 Mac OS X Developer Tools" pour Mac OS X 10.2 ou XCode pour Mac OS X 10.3. Les premiers sont disponibles sur le site Apple : "<http://connect.apple.com>". XCode est disponible avec les CDs Mac OS X 10.3 (voir en page À Savoir sur Blady pour l'installation Mac OS X). Pour utiliser la version 3.1 du compilateur il est alors nécessaire de le sélectionner :

```
$ sudo gcc_select 3.1
```

Télécharger les fichiers suivants :

"10.2 Compiler" -> "gnat-gcc-3.1-1173-jaguar.dmg.gz",
"gdb wavefront for 10.2" -> "gnatgdb-225.20020911.sit",
sur le site "<http://www.macada.org>" aux pages "Compilers" et "Debuggers".

Le compilateur :

Ouvrir "gnat-gcc-3.1-1173-jaguar.dmg.gz" puis "gnat-gcc-3.1-1173-jaguar.dmg" sous le Finder, et lancer l'installation du compilateur contenu dans "gcc_gnat_macosx.pkg".

Le compilateur s'installe dans les répertoires :

```
/usr/bin  
/usr/include/gcc/darwin/3.1  
/usr/lib/gcc/darwin/3.1  
/usr/libexec/gcc/darwin/ppc/3.1
```

Le dévermineur pour 10.2 (Attention, il n'est pas compatible avec XCode) :

Ouvrir "gnatgdb-225.20020911.sit", placer le contenu de l'archive sur votre bureau.

```
$ cd /usr/bin  
$ sudo mv gdb gdb-apple  
$ sudo mv ~/Desktop/gnatgdb-225.20020911 gdb  
$ sudo chmod +x gdb  
$ sudo chown root gdb  
$ sudo chgrp wheel gdb
```

2) Utilisation avec le Terminal

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

```
% gnatmake hello.adb
```

Le fichier hello.adb étant :

```
with Text_IO; use Text_IO;
procedure Hello is
begin
put_line("Hello again, avec Ada.");
end;
```

Et les résultats ne se font pas attendre :

```
% gnatmake hello.adb
gcc -c hello.adb
gnatbind -x hello.ali
gnatlink hello.ali
% ./hello
Hello again, avec Ada.
```

3) les commandes utiles avec le Terminal

La liste des commandes est obtenue de la façon suivante :

```
$ gnatcmd
```

```
List of available commands
```

GNAT BIND file[.ali] /qualifiers	gnatbind
GNAT CHOP file [directory] /qualifiers	gnatchop
GNAT COMPILE filespec[,...] /qualifiers	gcc -c -x ada
GNAT ELIM name /qualifiers	gnatelim
	déetecte et élimine les sous-programmes inutilisés
GNAT FIND pattern[:sourcefile[:line[:column]]] filespec[,...] /qualifiers	gnatfind
	liste toutes les utilisations d'une entité Ada
GNAT KRUNCH file [/COUNT=nnn]	gnatkr
GNAT LIBRARY [/CREATE SET DELETE]=directory [/CONFIG=file]	gnatlb
GNAT LINK file[.ali] [extra obj_&_lib_&_exe_&_opt files] /qualifiers	gnatlink

GNAT LIST /qualifiers object_or_ali_file	gnatls
liste le contenu des objets générés	
GNAT MAKE file /qualifiers (includes COMPILE /qualifiers)	gnatmake
utilitaire optimisé de compilation multi-unités	
GNAT PREPROCESS ifile ofile dfile /qualifiers	gnatprep
pré-processeur externe	
GNAT SHARED [obj_&_lib_&_exe_&_opt files] /qualifiers	
gcc -shared -L/usr/local/gnat/lib/gcc-lib/powerpc-apple-darwin5.2/3.1/adalib/ -lgnat	
GNAT STANDARD	gnatpsta
affiche le package "Standard"	
GNAT STUB file [directory] /qualifiers	gnatstub
générateur de "body" vierges mais compilables	
GNAT SYSTEM	gnatpsys
affiche le package "System"	
GNAT XREF filespec[,...] /qualifiers	gnatxref
utilitaire d'édition des références croisées	

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

% gnatmake (extrait)

Usage: gnatmake opts name {[-cargs opts] [-bargs opts] [-larges opts]}
 name is a file name from which you can omit the .adb or .ads suffix

gnatmake switches:

- b Bind only
- c Compile only
- f Force recompilations of non predefined units
- k Keep going after compilation errors
- l Link only
- M List object file dependences for Makefile
- v Display reasons for all (re)compilations

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
- larges opts opts are passed to the linker

Compiler switches (passed to the compiler by gnatmake):

- g Generate debugging information
- ldir Specify source files search path
- O[0123] Control the optimization level
- gnatf Full errors. Verbose details, all undefined references
- gnatv Verbose mode. Full error output with source lines to stdout
- gnat83 Enforce Ada 83 restrictions

Et aussi avec gcc :

```
% gcc --help  
...
```

5) Convention des noms de fichiers avec GNAT

'main.ads'

Main (spec), contient la spécification d'une procédure ou d'une fonction, utile pour l'appeler d'une autre unité.

'main.adb'

Main (body), contient le code de la procédure ou de la fonction principale du programme.

'arith_functions.ads'

Arith_Functions (package ou proc ou func spec), contient les spécifications d'une unité Ada.

'arith_functions.adb'

Arith_Functions (package ou proc ou func body), contient le code d'une unité Ada.

'func-spec.ads'

Func.Spec (child package spec), contient les spécifications d'une unité fille Ada.

'func-spec.adb'

Func.Spec (child package body), contient le code d'une unité fille Ada.

'main-sub.adb'

Sub (subunit of Main separate), contient le code d'une unité Ada définie comme séparée.

Attention, par défaut, le nom des fichiers doit être identique à celui des unités définies dans le source.

Pascal Pignard, novembre 2003.