

Installation GNAT OS X 10.1

1) Installation du compilateur (version gcc3-1041)

Télécharger les fichiers "gnat-gcc3.dmg.gz", "Carbon_Bindings.sit", "gnatgdb-208-020221.gz", sur le site "<http://www.adapower.net/macos>" aux pages "Compilers", "Bindings" et "Debuggers".

Le compilateur :

Ouvrir "gnat-gcc3.dmg.gz" sous le Finder, et lancer l'installation du compilateur contenu dans "gnat-gcc3.pkg".

Les interfaces de programmation Ada :

Ouvrir "Carbon_Bindings.sit" sous le Finder. Les APIs Carbon sont placées sous /usr/local/Bindings/Frameworks. Dans le même temps, les modèles (voir plus loin) pour Project Builder sont placés sous /Developer/ProjectBuilder Extras/Project Templates.

Le dévermineur :

```
% gunzip gnatgdb-208-020221.gz
% sudo install -c -m 775 gnatgdb-208-020221 /usr/local/gnat/bin/gdb
```

Le compilateur s'installe dans le répertoire /usr/local/gnat.

Pour pouvoir utiliser GNAT, il est indispensable d'avoir installé les derniers outils de développement "April2002DevTools", version gcc3-1041. Ils sont disponibles sur le site Apple : <http://developer.apple.com/tools/macosxtools.html>. (voir en page À Savoir sur Blady pour l'installation Mac OS X).

2) Utilisation avec le Terminal

Après s'être connecté sur son compte propre, l'ajout de la localisation du compilateur est nécessaire pour une première fois dans le fichier ~/.tcshrc par exemple :

```
setenv PATH /usr/local/gnat/bin:$PATH
setenv MANPATH /usr/local/gnat/man:$MANPATH
```


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

...

4) Utilisation avec ProjectBuilder

Lancer ProjectBuilder et choisir "New Project..." dans le menu "File". La fenêtre "New Project" s'affiche avec le choix entre trois projets Ada :

- Ada Carbon Application : permet de construire des applications Carbon avec une interface utilisateur construite à base de ressources "rcrs",
- Ada Carbon Application (Nib Based) : permet de construire des applications Carbon avec une interface utilisateur construite avec InterfaceBuilder.
- Ada Standard Tool : permet de construire des programmes utilisables sous Terminal.

Les applications se construisent alors de manière classique.

L'ajout de fichiers s'opère en choisissant "New File..." dans le menu "File". La fenêtre "New File" s'affiche avec le choix entre trois fichiers Ada :

- Ada Package : permet de construire le corps d'une unité Ada,
- Ada Spec : permet de construire la spécification d'une unité Ada,
- Ada Program : permet de construire la procédure principale d'un programme Ada.

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, janvier, février, mai, août, septembre 2002, janvier 2003.