

Installation GNAT OS X 10.11

Deux sources du compilateur GNAT existent pour MacOS 10.11 :

- le compilateur GNAT GPL basé sur GCC 4.9 (juin 2016), c'est le plus officiel,
- le compilateur GNAT FSF basé sur GCC 5.2 (novembre 2015), inclut les langages supplémentaires C++, Fortran et Objective C.

Sommaire

1.	Installation du compilateur GNAT GPL 2016 basé sur GCC 4.9	2
2.	Installation du compilateur GNAT FSF basé sur GCC 5.2	5
3.	Utilisation avec le Terminal	6
4.	Les commandes utiles avec le Terminal	6

1. Installation du compilateur GNAT GPL 2016 basé sur GCC 4.9

Télécharger le fichier suivant sur le bureau du Mac :

Plateforme "x86-64 Mac OS X (64 bits)" "GNAT GPL 2016" : "gnat-gpl-2016-x86_64-darwin-bin.tar.gz", sur le site "<https://libre.adacore.com>" à la page "Download GNAT GPL" puis cocher "Free Software or Academic Development" et cliquer sur "Build your download package".

Lancer le Terminal dans une session administrateur et taper les commandes suivantes (réponses à fournir en gras) :

```
# Créer le répertoire /usr/local/gnat (si pas déjà fait)
$ cd /usr/local/gnat
$ tar xzf ~/Desktop/gnat-gpl-2016-x86_64-darwin-bin.tar.gz
$ cd gnat-gpl-2016-x86_64-darwin-bin
$ ./doinstall
...
Please press RETURN to continue.<ret>
...
In which directory do you want to install GNAT? [/usr/local/gnat]:<ret>
...
Do you want to continue ? [yYlnN]:y<ret>
...
Do you want to proceed with installation ? [yYlnN]:y<ret>
...
On OS X, the debugger needs to be codesigned in order to access
the priviledges required for debugging.

Please refer to the procedure documented in the Mac OS Topics chapter
of the GNAT GPL User's Guide, section Codesigning the Debugger. This
procedure should be applied after the GNAT GPL install has completed.

Press Enter to continue.<ret>
...
GNAT GPL is now installed.
...
Thank you for installing GNAT GPL Edition!
```

Tel qu'il est indiqué, nous devons alors signer GDB pour qu'il fonctionne avec MacOS.

a) Ouvrir l'application "Trousseau d'accès" dans le Dock->LauchPad->Autre.
Sélectionner le menu "Trousseau d'accès->Assitant de certification->Créer un certificat...".

Dans la fenêtre qui apparait :

- . donner le nom "gdb-2016",
- . le type d'identité à "Racine auto-signée",
- . le type de certificat à "Signature de code",
- . cocher le case "Me laisser ignorer les réglages pas défaut".

Puis cliquer sur le bouton Continuer plusieurs fois jusqu'à ce qu'apparaisse le panneau "Indiquez l'emplacement du certificat."

b) Sélectionner alors "Système", puis sur le bouton Créer. Une fenêtre d'autorisation de modification du trousseau apparaît, entrer le mot de passe puis cliquer sur le bouton "Modifier le trousseau". Le certificat est créé, cliquer alors sur le bouton Terminer.

c) Dans la fenêtre des trousseau, sélectionner le trousseau Système et double-cliquer sur le certificat gdb-2016. Dans la fenêtre qui apparaît, déployer le triangle "Se fier" et en face de "Lors de l'utilisation de ce certificat" sélectionner "Toujours approuver" puis fermer la fenêtre en cliquant sur sa bulle rouge de fermeture. Une fenêtre d'autorisation de modification des réglages apparaît, entrer le mot de passe puis cliquer sur le bouton "Mettre à jour les réglages".

Quitter l'application, ça été long et ce n'est pas tout à fait fini. Il nous faut alors redémarrer le Mac.

Nous pouvons alors enfin signer GDB :

```
$ cd /usr/local/gnat/bin
$ codesign -f -s "gdb-2016" gdb
gdb: replacing existing signature
```

Comme il bien recommandé, utiliser à chaque fois la commande suivante pour une utilisation temporaire :

```
$ PATH=/usr/local/gnat/bin:$PATH
$ export MANPATH=/usr/local/gnat/share/man:$MANPATH
```

Pour une utilisation courante, saisir aussi les commandes suivantes :

```
$ echo 'PATH=/usr/local/gnat/bin:$PATH' >> ~/.profile
$ echo 'PATH=/usr/local/gnat/bin:$PATH' >> ~/.bashrc
$ echo 'export MANPATH=/usr/local/gnat/share/man:$MANPATH' >> ~/.profile
$ echo 'export MANPATH=/usr/local/gnat/share/man:$MANPATH' >> ~/.bashrc
```

Une documentation aux formats HTML, PDF, TXT et "info" est disponible dans le répertoire /usr/local/gnat/share/doc/gnat.

```
$ open /usr/local/gnat/share/doc/gnat/html/arm12.html
$ open /usr/local/gnat/share/doc/gnat/html/gcc.html
$ open /usr/local/gnat/share/doc/gnat/html/gdb.html
$ open /usr/local/gnat/share/doc/gnat/html/gnat_rm.html
$ open /usr/local/gnat/share/doc/gnat/html/gnat_ugn.html
$ open /usr/local/gnat/share/doc/gnat/html/gnatcheck_rm.html
```

Des exemples de programmes sont disponibles dans le répertoire /usr/local/gnat/share/examples.

Correctif de gnatdoc :

L'utilitaire de génération de documentation n'est pas fonctionnel :

```
$ gnatdoc
dyld: Library not loaded: @rpath/libclang.dylib
  Referenced from: /usr/local/gnat/bin/gnatdoc
  Reason: image not found
Trace/BPT trap: 5
```

La bibliothèque libclang.dylib n'est pas référencée correctement, nous allons effectuer la correction avec la commande suivante :

```
$ cd /usr/local/gnat/bin
$ install_name_tool -change @rpath/libclang.dylib @loader_path/../lib/gps/libclang.dylib
gnatdoc
```

Correctif de GPS :

Au lancement, GPS averti qu'il manque des données :

```
[2016-07-10 18:33:40] /usr/local/gnat-2016/share/doc/gprbuild/html/gps_index.xml:1:0: No
root element specified
```

Ce fichier est vide, nous allons effectuer la correction en ajoutant les lignes suivantes dans le fichier :

```
<?xml version="1.0" ?>
<!-- This file makes the gprbuild documentation available from GPS -->
<GPS>

<submenu before="About">
  <title>/Help/Gprbuild</title>
</submenu>

<documentation_file>
  <name>gprbuild_ug.html</name>
  <descr>Gprbuild User's Guide</descr>
  <category>GNAT</category>
  <menu>/Help/Gprbuild/Gprbuild User's Guide</menu>
</documentation_file>

</GPS>
```

L'erreur disparaît et le menu d'aide de gprbuild est maintenant actif dans GPS.

2. Installation du compilateur GNAT FSF basé sur GCC 5.2

Télécharger le fichier suivant sur le bureau du Mac :

Compilateur FSF gcc 5.2 : "gcc-5.2.0-x86_64-apple-darwin15-bin.tar.bz2",
sur le site GNUAda "https://sourceforge.net/projects/gnuada/files/GNAT_GCC%20Mac%20OS%20X/5.2.0/native".

(Les instructions de construction du compilateur avec les langages supplémentaires Fortran et Objective-C sont sur :

<http://forward-in-code.blogspot.co.uk/2015/04/building-gcc-510.html>)

Lancer le Terminal dans une session administrateur et taper les commandes suivantes :

```
# Créer le répertoire /opt/gcc-5.2.0
$ cd /opt/gcc-5.2.0
$ tar jxvf ~/Desktop/gcc-5.2.0-x86_64-apple-darwin15-bin.tar.bz2
$ cd gcc-5.2.0-x86_64-apple-darwin15-bin
$ sudo ./doinstall
...
Please press RETURN to continue.<ret>
...
In which directory do you want to install GNAT? [/usr/local/gnat]:<ret>
...
Do you want to continue ? [yYlnN]:y<ret>
...
Do you want to proceed with installation ? [yYlnN]:y<ret>
...
On OS X, the debugger needs to be codesigned in order to access
the priviledges required for debugging.

Please refer to the procedure documented in the Mac OS Topics chapter
of the GNAT GPL User's Guide, section Codesigning the Debugger. This
procedure should be applied after the GNAT GPL install has completed.

Press Enter to continue.
...
GNAT GPL is now installed.
...
Thank you for installing GNAT GPL Edition!
```

Tel qu'il est indiqué, nous devons alors signer GDB pour qu'il fonctionne avec MacOS.

Le compilateur s'installe à partir du répertoire :
/opt/gcc-4.8.0

Pour une utilisation courante, saisir aussi les commandes suivantes :

```
$ echo 'PATH=/opt/gcc-4.8.0/bin:$PATH' >> ~/.profile
$ echo 'PATH=/opt/gcc-4.8.0/bin:$PATH' >> ~/.bashrc
```

Pour une utilisation temporaire, utiliser à chaque fois la commande suivante :

```
$ export PATH=/opt/gcc-4.8.0/bin:$PATH
```

Une documentation aux formats "info" et "man" est disponible dans les répertoires /opt/gcc-4.8.0/share/info et /opt/gcc-4.8.0/share/man.

```
$ info -f /opt/gcc-4.8.0/share/info/dir
```

```
$ man -M /opt/gcc-4.8.0/share/man gcc
```

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

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.
```

4. Les commandes utiles avec le Terminal

La liste des commandes est obtenue de la façon suivante :

```
$ gnat
GNAT GPL 2016 (20160515-49)
Copyright 1996-2016, Free Software Foundation, Inc.
List of available commands
GNAT BIND          gnatbind
réalise l'édition des liens des unités Ada compilées
GNAT CHOP          gnatchop
découpe un fichier en unités pour satisfaire les conventions Gnat
GNAT CLEAN         gnatclean
```

nettoie les fichiers générés par gnat
GNAT COMPILE gnatmake -f -u -c
compile une entité Ada
GNAT CHECK gnatcheck
vérifie le code source suivant des règles définies (non présent avec gnat-osx)
GNAT ELIM gnatelim
détecte et élimine les sous-programmes inutilisés
GNAT FIND gnatfind
liste toutes les utilisations d'une entité Ada
GNAT KRUNCH gnatkr
réduit les noms de fichiers au nombre maximal de lettres spécifié
GNAT LINK gnatlink
réalise l'édition des liens de l'exécutable
GNAT LIST gnatls
liste le contenu des objets générés
GNAT MAKE gnatmake
utilitaire optimisé de compilation multi-unités
GNAT METRIC gnatmetric
statistiques sur le code Ada
GNAT NAME gnatname
réalise la correspondance entre les unités Ada et les noms des fichiers lorsque ceux-ci ne
sont pas au standard GNAT
GNAT PREPROCESS gnatprep
pré-processeur externe
GNAT PRETTY gnatpp
reformate le source Ada
GNAT STACK gnatstack
calcul la taille de pile mémoire maximale théorique (non présent avec gnat-gpl)
GNAT STUB gnatstub
créé le squelette d'un corps d'une spécification
GNAT TEST gnattest
créé ou exécute la suite de test unitaire
GNAT XREF gnatxref
utilitaire d'édition des références croisées
De même chacune des commandes ci-dessus 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:

- version Display version and exit
- help Display usage and exit
- c Compile only
- f Force recompilations of non predefined units
- k Keep going after compilation errors
- 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
- s Recompile if compiler switches have changed

- u Unique compilation. Only compile the given file.
- v Display reasons for all (re)compilations
- z No main subprogram (zero main)

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

Compiler switches (passed to the compiler by gnatmake):

- ldir Specify source files search path
- gnat83 Ada 83 mode
- gnat95 Ada 95 mode
- gnat2005 Ada 2005 mode
- gnat2012 Ada 2012 mode (default)

Et aussi avec gcc :

```
$ gcc --help
```

...

Pour en savoir plus : voir l'utilisation avancée des outils GNAT GPL 2016 avec le compilateur GCC 4.9.4, l'environnement de développement GPS 2016 ainsi que du débogueur GDB 7.10 et des exceptions Ada sur la page à savoir de Blady.

Pascal Pignard, juin 2014, juin, décembre 2015, juin, août 2016.