

Installation de GNAT 2020 pour macOS 10.15

Deux sources du compilateur GNAT existent pour macOS 10.15 :

- le compilateur GNAT Community 2020 basé sur GCC 8.4.1 (août 2020) pour les langages Ada, C et C++, c'est le plus officiel,
- le compilateur GNAT FSF basé sur GCC 9.1.0 (août 2019), inclut les langages supplémentaires Fortran et Objective C.

Sommaire

1.	Installation du compilateur GNAT Community 2020 basé sur GCC 8.4.1	2
2.	Installation du compilateur GNAT FSF basé sur GCC 9.1.0	4
3.	Utilisation avec le Terminal	6
4.	Les commandes utiles avec le Terminal	6

1. Installation du compilateur GNAT Community 2020 basé sur GCC 8.4.1

Télécharger le fichier suivant sur le bureau du Mac :

À partir du site www.adacore.com/download à la page "More packages, platforms, versions and sources" sélectionner la plateforme "x86 Mac OS X (64 bits)" "2020" :
"gnat-2020-20200818-x86_64-darwin-bin.dmg",

Depuis votre session administrateur, ouvrir le fichier téléchargé puis lancer l'application d'installation "gnat-2020-20200818-x86_64-darwin-bin.app" (clic droit puis Ouvrir deux fois pour autoriser explicitement l'ouverture d'un programme provenant d'Internet). Cliquer sur le bouton "Next" puis sélectionner le dossier destination. Vérifier que vous êtes bien le propriétaire du dossier sélectionné (ce qui n'est pas forcément le cas avec par exemple /usr/local/gnat) sinon l'installation pourrait se bloquer. Puis cliquer sur le bouton "Next".

Vous pouvez alors sélectionner les composants à installer, pour ma part je laisse tel quel (tous par défaut), cliquer sur le bouton "Next". L'acceptation de la licence GPL V3 apparaît, cliquer sur le bouton "Next". Il ne reste plus qu'à cliquer sur le bouton "Install" pour lancer l'installation.

Après un peu d'attente - soyez patient, évitez de cliquer sur la vue détail car cela peut provoquer un blocage - le compilateur GNAT et le prouveur formel SPARK sont installés, cliquer sur le bouton "Finish".

L'éditeur intégré GPS ne fait plus partie de cette livraison, cependant celui de la livraison de 2019 est utilisable avec GNAT 2020, à installer dans un autre dossier (voir sur Blady) puis déclarer l'alias suivant :

```
% alias gps=/usr/local/adacore/2019/bin/gps
```

C'est presque terminé, comme il bien recommandé, utiliser à chaque fois la commande suivante pour une utilisation temporaire du compilateur :

```
$ 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
```

D'autre part, le debugger gdb n'est pas encore fonctionnel, il est bloqué par le système de surveillance de macOS et provoque cette erreur à l'exécution du programme à débbuger :

```
Unable to find Mach task port for process-id 6633: (os/kern) failure (0x5).  
(please check gdb is codesigned - see taskgated(8))
```

Nous allons suivre la procédure décrite sur le blog de Simon Wright pour qu'il fonctionne avec macOS (forward-in-code.blogspot.com/2018/11/mojave-vs-gdb.html). Lancer le Terminal dans une session administrateur et taper les commandes suivantes :

a) Ouvrir l'application "Trousseau d'accès" dans le dossier "Applications -> Utilitaires". Sélectionner le menu "Trousseau d'accès -> Assistant de certification -> Créer un certificat...".

Dans la fenêtre qui apparaît :

- . donner le nom "gdb-cert",
- . 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 trousseaux, sélectionner le trousseau Système et double-cliquer sur le certificat gdb-cert. 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 Trousseau, ça été long et ce n'est pas tout à fait fini. Il nous faut alors redémarrer le Mac (c'est malheureusement nécessaire) puis télécharger le fichier de description gdb.xml à partir de Blady :

blady.pagesperso-orange.fr/telechargements/gnat/gdb.xml

Nous pouvons alors enfin signer GDB :

```
$ cd /usr/local/gnat/bin
$ codesign --force --sign gdb-cert --entitlements ~/Downloads/gdb.xml gdb
Password:
```

Une fenêtre s'ouvre "macOS souhaite effectuer des modifications", saisir nom administrateur et mot de passe puis cliquer sur le bouton "Autoriser" pour autoriser la signature.

Une documentation aux formats HTML, PDF, TXT et "info" est disponible dans le dossier `/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/gnat_rm.html
$ open /usr/local/gnat/share/doc/gnat/html/gnat_ugn/gnat_ugn.html
```

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

2. Installation du compilateur GNAT FSF basé sur GCC 9.1.0

Télécharger le fichier suivant sur le bureau du Mac :

Compilateur FSF gcc 9.1 : "gcc-9.1.0-x86_64-apple-darwin15-bin.tar.bz2",
sur le site GNUAda sourceforge.net/projects/gnuada/files/

[GNAT_GCC%20Mac%20OS%20X/9.1.0/native](https://sourceforge.net/projects/gnuada/files/GNAT_GCC%20Mac%20OS%20X/9.1.0/native).

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

forward-in-code.blogspot.co.uk/2015/04/building-gcc-510.html

Et un ensemble de scripts pour aider la construction sont sur :

github.com/simonjwright/building-x86_64)

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

```
$ cd /opt
$ sudo mkdir gcc-9.1.0
$ sudo chown $USER gcc-9.1.0
$ cd gcc-9.1.0
$ tar jxf ~/Desktop/gcc-9.1.0-x86_64-apple-darwin15-bin.tar.bz2
$ cd gcc-9.1.0-x86_64-apple-darwin15-bin
$ sudo ./doinstall
```

...

Please press RETURN to continue.<ret>

...

In which directory do you want to install FSF GCC 9.1.0?
[/opt/gcc-9.1.0]:<ret>

...

Do you want to continue ? [yYlnN]:y<ret>

...

Do you want to proceed with installation ? [yYlnN]:y<ret>

...

On macOS, the debugger needs to be codesigned in order to access the privileges required for debugging.

If running on macOS prior to Mojave (10.14), please refer to the procedure documented in the Mac OS Topics chapter of the GNAT User's Guide, section Codesigning the Debugger, at

http://docs.adacore.com/gnat_ugn-docs/html/gnat_ugn/gnat_ugn.html.

If running on Mojave, please refer to the updated procedure documented at <https://forward-in-code.blogspot.com/2018/11/mojave-vs-gdb.html>, section "Security".

These procedures should be applied after the FSF GCC install has completed.

Press Enter to continue.<ret>

...

```
FSF GCC 9.1.0 is now installed. To launch it, you must put
/opt/gcc-9.1.0/bin
in front of your PATH environment variable. The following
commands enable you to do this:
  PATH=/opt/gcc-9.1.0/bin:$PATH; export PATH (Bourne shell)
  setenv PATH /opt/gcc-9.1.0/bin:$PATH (C shell)
Thank you for installing FSF GCC 9.1.0!
```

Tel qu'il est indiqué, nous devons alors signer GDB pour qu'il fonctionne avec macOS. Il nous faut auparavant supprimer la mise en quarantaine par Catalina :

```
$ cd /opt
$ sudo xattr -r -d com.apple.quarantine gcc-9.1.0
```

Suivre alors la procédure du §1 pour signer gdb. Ce GDB a aussi besoin pour toutes ses utilisations d'une option à désactiver que l'on mettra alors dans le fichier .gdbinit du dossier \$HOME de sa session :

```
$ cat >> $HOME/.gdbinit
set startup-with-shell off
<ctrl>D
```

Le compilateur s'installe à partir du dossier : /opt/gcc-9.1.0

Pour une utilisation courante, saisir aussi les commandes suivantes :

```
$ echo 'PATH=/opt/gcc-9.1.0/bin:$PATH' >> ~/.profile
$ echo 'PATH=/opt/gcc-9.1.0/bin:$PATH' >> ~/.bashrc
```

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

```
$ export PATH=/opt/gcc-9.1.0/bin:$PATH
```

Une documentation aux formats "info" et "man" est disponible dans les répertoires /opt/gcc-9.1.0/share/info et /opt/gcc-9.1.0/share/man.

```
$ info -f /opt/gcc-9.1.0/share/info/dir
$ man -M /opt/gcc-9.1.0/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 Community 2020 (20200818-84)
Copyright 1996-2020, Free Software Foundation, Inc.
List of available commands
GNAT BIND          gnatbind      réalise l'édition des liens des unités Ada compilées
GNAT CHOP          gnat chop     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        gnat check   vérifie le code source suivant des règles définies (non
présent avec gnat-osx)
GNAT ELIM         gnat elim    détecte et élimine les sous-programmes inutilisés
GNAT FIND         gnat find    liste toutes les utilisations d'une entité Ada
GNAT KRUNCH       gnat kr      réduit les noms de fichiers au nombre maximal de lettres
spécifié
GNAT LINK         gnat link    réalise l'édition des liens de l'exécutable
GNAT LIST         gnat ls      liste le contenu des objets générés
GNAT MAKE         gnat make    utilitaire optimisé de compilation multi-unités
GNAT METRIC       gnat metric  statistiques sur le code Ada
GNAT NAME         gnat name    réalise la correspondance entre les unités Ada et les
noms des fichiers lorsque ceux-ci ne sont pas au standard GNAT
GNAT PREPROCESS   gnat prep    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	gnatstest	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
- larges opts opts are passed to the linker

Compiler switches (passed to the compiler by gnatmake):

- ldir Specify source files search path
- gnat2012 Ada 2012 mode (default)

Et aussi avec gcc :

\$ gcc --help

...

Pour en savoir plus : voir l'utilisation avancée des outils GNAT GPL 2020 avec le compilateur GCC 8.4.1, l'environnement de développement GPS 2020 ainsi que du dévermineur GDB 7.10 et des exceptions Ada sur la page à savoir de Blady.

Pascal Pignard, juin 2018, mai 2019, janvier 2020, février 2020, octobre 2020.