

Installation de GNAT 2021 pour macOS 11

Deux sources du compilateur GNAT existent pour macOS 11 :

- le compilateur GNAT Community Edition 2021 basé sur GCC 10.3.1 (mai 2021) 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 CE 2021 basé sur GCC 10.3.1	2
2.	Construction du compilateur GNAT CE 2021 basé sur GCC 10.3.1	5
3.	Installation du compilateur GNAT FSF basé sur GCC 9.1.0	11
4.	Utilisation avec le Terminal	13
5.	Les commandes utiles avec le Terminal	13

1. Installation du compilateur GNAT CE 2021 basé sur GCC 10.3.1

Télécharger le fichier suivant dans votre dossier Téléchargements :

Compilateur GNAT GPL 2021-x86_64-darwin-bin : "gnat-ce-2021-x86_64-apple-darwin15.pkg", depuis le site GNUAda :
sourceforge.net/projects/gnuada/files/GNAT_GPL%20Mac%20OS%20X/2021-x86_64-darwin-bin

Lancer l'application d'installation "gnat-ce-2021-x86_64-apple-darwin15.pkg" (clic droit puis Ouvrir pour autoriser explicitement l'ouverture d'un programme provenant d'Internet). Cliquer sur le bouton "Continuer" trois fois puis acquiescer la licence avec un clic sur le bouton "Agree". Changer le dossier destination si vous n'êtes pas satisfait avec le dossier par défaut : "/opt/gnat-ce-2021". Puis cliquer sur le bouton "Installer".

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. Une fenêtre apparaît, saisir un utilisateur administrateur et son mot de passe. Le résumé de l'installation apparaît, cliquer sur le bouton "Fermer", une fenêtre apparaît, cliquer sur le bouton "Conserver" pour garder le programme d'installation.

L'éditeur intégré GPS ne fait pas partie de cette livraison, cependant celui de la livraison de 2019 est utilisable avec GNAT CE 2021, à installer dans un autre dossier (voir sur Blady) puis déclarer l'alias suivant avec la commande suivante, par exemple :

```
% 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=/opt/gnat-ce-2021/bin:$PATH  
$ export MANPATH=/opt/gnat-ce-2021/share/man:$MANPATH
```

Pour une utilisation courante, saisir aussi les commandes suivantes :

```
$ echo 'PATH=/opt/gnat-ce-2021/bin:$PATH' >> ~/.zprofile  
$ echo 'PATH=/opt/gnat-ce-2021/bin:$PATH' >> ~/.zshrc  
$ echo 'export MANPATH=/usr/local/gnat/share/man:$MANPATH' >> ~/.zprofile  
$ echo 'export MANPATH=/usr/local/gnat/share/man:$MANPATH' >> ~/.zshrc
```

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 déboguer :
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 /opt/gnat-ce-2021/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 "/opt/gnat-ce-2021/share/doc/gnat".

```
$ open /opt/gnat-ce-2021/share/doc/gnat/html/arm12.html  
$ open /opt/gnat-ce-2021/share/doc/gnat/html/gcc.html  
$ open /opt/gnat-ce-2021/share/doc/gnat/html/gdb.html  
$ open /opt/gnat-ce-2021/share/doc/gnat/html/gnat_rm/gnat_rm.html  
$ open /opt/gnat-ce-2021/share/doc/gnat/html/gnat_ugn/gnat_ugn.html
```

Des exemples de programmes sont disponibles dans le dossier "/opt/gnat-ce-2021/share/examples".

2. Construction du compilateur GNAT CE 2021 basé sur GCC 10.3.1

GCC est le compilateur libre de la FSF bien connu pour être multi-plateformes et multi-cibles. Il prends en compte également de multiples langages informatiques : C, C++, Objective-C, Go, D, Fortran et bien sûr Ada. La partie Ada est maintenue avec AdaCore pour former l'ensemble des outils GNAT.

Nous allons construire l'ensemble ainsi que le débogueur GDB et les outils de manipulation des projets GPR.

NB : cette construction est légèrement différente de celle qui a donné lieu à la distribution installée au \$1. La construction présentée ici est réalisée sur macOS 11 alors que celle du \$1 est réalisée sur macOS 10.11. Cette dernière construction est décrite ici : github.com/simonjwright/building-gcc-macos-native.

La livraison de GNAT CE 2021 n'inclut pas de distribution pour macOS. Nous allons alors utiliser les sources fournies dans la distribution pour Linux (64 bits).

À partir du site GNAT Community Edition aller à la page "More packages, platforms, versions and sources", sélectionner la plateforme "x86 GNU Linux (64 bits)" "2021" et ouvrir la liste "Sources".

Télécharger les fichiers suivants dans votre dossier Téléchargements :

- gcc-10-2021-20210519-19A74-src.tar.gz
- gcc-interface-10-2021-20210519-19A75-src.tar.gz
- gdb-10-2021-20210518-19C88-src.tar.gz
- gnat-2021-20210519-19A70-src.tar.gz
- gprbuild-2021-20210519-19A34-src.tar.gz
- mpc-1.0.3.tar.gz
- mpfr-3.1.5.tar.bz2
- xmlada-2021-20210518-19D50-src.tar.gz

La version fournie de la bibliothèque GMP n'est pas compatible avec macOS 11, nous prenons la dernière version en date.

Télécharger le fichier suivant dans votre dossier Téléchargements :

- gmp-6.2.1.tar.lz

Nous préparons notre environnement. Saisir les commandes suivantes dans le Terminal :

```
% inst_base=/usr/local
% cd $inst_base
% sudo mkdir src-2021
% sudo chown $USER src-2021
% src_gnat=$PWD/src-2021
% sudo mkdir gnat-2021
% sudo chown $USER gnat-2021
% inst_gnat=$PWD/gnat-2021
```

a) Construire GMP

GNU Multiple Precision (GMP) est une bibliothèque de calcul arithmétique multi-précision sur des nombres entiers, rationnels et flottants.

Site web : gmplib.org.

Version installée : 6.2.1.

Dépendances : sans.

Pour la construction de GMP nous utiliserons le compilateur natif du Mac, adapter la variable PATH en conséquence :

```
% which gcc
/usr/bin/gcc
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $src_gnat
% tar xzf ~/Downloads/gmp-6.2.1.tar.lz
% cd gmp-6.2.1
% ./configure --prefix=$inst_gnat
% make
% make check
% make install
```

b) Construire MPFR

Multiple Precision FR (MPFR) est une bibliothèque de calcul arithmétique multi-précision sur des nombres flottants avec arrondi correct.

Site web : www.mpfr.org.

Version installée : 3.1.5.

Dépendances : GMP.

Pour la construction de MPFR nous utiliserons le compilateur natif du Mac, adapter la variable PATH en conséquence :

```
% which gcc
/usr/bin/gcc
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $src_gnat
% tar xzf ~/Downloads/mpfr-3.1.5.tar.bz2
% cd mpfr-3.1.5
% ./configure --prefix=$inst_gnat --with_gmp=$inst_gnat
% make
% make check
% make install
```

c) Construire MPC

Multiple Precision Complex (MPC) est une bibliothèque de calcul arithmétique multi-précision sur des nombres flottants complexes avec arrondi correct.

Site web : www.multiprecision.org/mpc.

Version installée : 3.1.5.

Dépendances : GMP, MPFR.

Pour la construction de MPC nous utiliserons le compilateur natif du Mac, adapter la variable PATH en conséquence :

```
% which gcc
/usr/bin/gcc
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $src_gnat
% tar xzf ~/Downloads/mpc-1.0.3.tar.gz
% cd mpc-1.0.3
% ./configure --prefix=$inst_gnat --with-gmp=$inst_gnat --with-mpfr=$inst_gnat
% make
% make check
% make install
```

d) Construire GCC / GNAT

GNU Compiler Collection/ GNU NYU Ada Translator (GCC / GNAT) est un compilateur pour le langage Ada mais aussi C et C++.

Site web : gcc.gnu.org.

Version installée : 10.3.1.

Dépendances : GMP, MPFR, MPC.

Pour la construction de GCC / GNAT nous utiliserons le compilateur GNAT CE 2020 (voir son installation sur Blady), adapter la variable PATH en conséquence :

```
% which gcc
/usr/local/adacore/20200818/bin/gcc
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $src_gnat
% tar xzv ~/Downloads/gnat-2021-20210519-19A70-src.tar.gz
% tar xzf ~/Downloads/gcc-interface-10-2021-20210519-19A75-src.tar.gz
% tar xzf ~/Downloads/gcc-10-2021-20210519-19A74-src.tar.gz
% ln -s gcc-10-2021-20210519-19A74-src gcc-src
% cd gcc-src/gcc
% ln -s ../../gnat-2021-20210519-19A70-src/src/ada .
% cd ada
% ln -s ../../../gcc-interface-10-2021-20210519-19A75-src gcc-interface
```

```

% cd $src_gnat
% mkdir gcc-build
% cd gcc-build
% target=x86_64-apple-darwin20
# for full bootstrap build
% GCC_BOOT_LDFLAGS="-static-libstdc++ -static-libgcc -Wl,-
headerpad_max_install_names"
# for --disable-bootstrap build
% GCC_STAGE1_LDFLAGS="-static-libstdc++ -static-libgcc -Wl,-
headerpad_max_install_names"
% ./src/configure \
--prefix=$inst_gnat \
--without-libiconv-prefix \
--disable-libmudflap \
--disable-libstdcxx-pch \
--disable-lsanitizer \
--disable-libcc1 \
--disable-libcilkrts \
--disable-multilib \
--disable-nls \
--enable-languages=c,c++,ada \
--host=$target \
--target=$target \
--build=$target \
--with-gmp=$inst_gnat \
--with-mpfr=$inst_gnat \
--with-mpc=$inst_gnat \
--with-boot-ldflags="$GCC_BOOT_LDFLAGS" \
--with-stage1-ldflags="$GCC_STAGE1_LDFLAGS" \
--with-sysroot=/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/
Developer/SDKs/MacOSX.sdk \
--with-build-config=no
% make
% make install

```

e) Construire GDB

GNU Debugger (GDB) est un dévermineur pour le langage Ada mais aussi C et C++.

Site web : www.gnu.org/software/gdb.

Version installée : 10.2.90.

Dépendances : sans

Pour la construction de GDB nous utiliserons le compilateur GCC tout juste construit ci-dessus, adapter la variable PATH en conséquence :

```

% which gcc
/opt/gnat-ce-2021-2021/bin/gcc

```


Saisir les commandes suivantes dans le Terminal :

```
% cd $src_gnat
% tar xzf ~/Downloads/gdb-10-2021-20210518-19C88-src.tar.gz
% ln -s gdb-10-2021-20210518-19C88-src gdb-src
% mkdir gdb-obj
% cd gdb-obj
% ../gdb-src/configure --build=$target --prefix=$inst_gnat --disable-werror
% make all
% cd gdb
% make install
```

Le débogueur gdb n'est pas encore fonctionnel, il est bloqué par le système de surveillance de macOS, nous le devons le signer suivant la procédure du § 1.

f) Construire XMLAda (Schema, DOM, SAX, Unicode)

XMLAda est une boîte à outil pour analyser les fichiers XML.

Site web : github.com/AdaCore/xmlada.

La version installée est CE 2021.

Dépendance : sans.

Avant de démarrer la compilation, le compilateur GNAT CE 2021 et les outils GPR de GNAT 2020 doivent être activés, adapter la variable PATH en conséquence :

```
% which gnat
/opt/gnat-ce-2021-2021/bin/gnat
% which gprbuild
/usr/local/adacore/20200818/bin/gcc
```

Récupérer ensuite, dans votre dossier Téléchargements, le fichier Makefile à partir du site :

raw.githubusercontent.com/Blady-Com/gnat-builder/xnadalib-2021/Makefile

Saisir les commandes suivantes dans le Terminal :

```
% cd $src_gnat
% cp -p ~/Downloads/Makefile .
% tar xzf ~/Downloads/xmlada-2021-20210518-19D50-src.tar.gz
% ln -s xmlada-2021-20210518-19D50-src xmlada-src
% make os=darwin prefix=$inst_gnat xmlada-options=--enable-build=Debug
GPRBUILD_OPTIONS="-gnatwn -gnatU" BUILDS_SHARED=yes xmlada
# Utilisation de Sphinx et LaTeX avec MacPorts (voir installation sur Blady)
% PATH=/opt/local/bin:$PATH make -C xmlada-build docs
% make os=darwin sudo= BUILDS_SHARED=yes xmlada-install
```

g) Construire les outils GPRBuild

GPRBuild est le moteur de construction d'un programme à travers ses sources en enchaînant les étapes de compilation, l'édition des liens Ada, l'édition des liens des binaires.

Un programme complexe peut être décrit à travers un fichier projet GPR. La bibliothèque de manipulation des fichiers GPR est aussi installée.

Site web : github.com/AdaCore/gprbuild.

La version installée est CE 2021.

Dépendance : XMLAda.

Avant de démarrer la compilation, le compilateur GNAT CE 2021 et les outils GPR de GNAT 2020 doivent être activés, adapter la variable PATH en conséquence :

```
% which gnat
/opt/gnat-ce-2021-2021/bin/gnat
% which gprbuild
/usr/local/adacore/20200818/bin/gcc
```

Le dossier d'installation de XMLAda doit être activé, ajouter son emplacement comme par exemple :

```
% export GPR_PROJECT_PATH=$inst_gnat/share/gpr
```

Récupérer ensuite, dans votre dossier Téléchargements, le fichier Makefile à partir du site :

raw.githubusercontent.com/Blady-Com/gnat-builder/xnadalib-2021/Makefile

Saisir les commandes suivantes dans le Terminal :

```
% cd $src_gnat
% cp -p ~/Downloads/Makefile .
% tar xzf ~/Downloads/gprbuild-2021-20210519-19A34-src.tar.gz
% ln -s gprbuild-2021-20210519-19A34-src gprbuild-src
% make os=darwin prefix=$inst_gnat gprbuild-options=BUILD=debug
GPRBUILD_OPTIONS="-gnatwn -gnatU" gprbuild
# Utilisation de Sphinx et LaTeX avec MacPorts (voir installation sur Blady)
% PATH=/opt/local/bin:$PATH make -C gprbuild-build doc
# La doc PDF n'est pas générée suite à des erreurs
% make os=darwin sudo= gprbuild-install
```

3. Installation du compilateur GNAT FSF basé sur GCC 9.1.0

Télécharger le fichier suivant dans votre dossier Téléchargements :
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
```

4. 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.
```

5. Les commandes utiles avec le Terminal

La liste des commandes est obtenue de la façon suivante :

```
$ gnat
GNAT Community 2021 (20210519)
Copyright 1996-2021, 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        gnatelim     détecte et élimine les sous-programmes inutilisés
GNAT FIND        gnat find    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 présent avec gnat-gpl)	gnatstack	calcul la taille de pile mémoire maximale théorique (non
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] [-largs 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
- 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
- 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
- largs opts opts are passed to the linker
- margs opts opts are passed to gnatmake

Compiler switches (passed to the compiler by gnatmake):

- ldir Specify source files search path
- gnat2012 Ada 2012 mode (default)
- gnat2022 Ada 2022 mode

Et aussi avec gcc :

\$ gcc --help

...

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

Pascal Pignard, juin 2018, mai 2019, janvier 2020, février 2020, octobre 2020, juillet 2021.