

# Installer ou construire GTKAda Quartz et autres composants en Ada sur macOS avec la bibliothèque XNAdaLib (Version CE 2021)

GTKAda est la boîte à outil graphique en Ada 95 basée sur GTK pour construire des applications portables sur la plupart des plateformes.

GTK est une bibliothèque graphique Linux conçue à l'origine pour fonctionner sur les systèmes Unix avec X-Windows mais les développeurs GTK ont inclus une interface de macOS utilisant le moteur graphique Quartz. Un programme peut donc être conçu pour tourner sur macOS sans activer le sous-système X11.

Site [www.gtk.org/download/macos.php](http://www.gtk.org/download/macos.php).

Nous pouvons soit construire GTKAda à partir des sources (voir §3 et suivants) soit le prendre prêt à l'emploi sur Source Forge (voir §2).

GTKAda est incluse dans la bibliothèque XNAdaLib qui comprend également le constructeur graphique d'application Glade ainsi que d'autres composants en Ada : Template Parser, Gate3, Simple Components, AICWL.

Nous allons voir comment installer la bibliothèque XNAdaLib prête à l'emploi et aussi comment la construire à partir des sources de ses composants.

## Sommaire

1.	Configuration	3
2.	Installer <i>GTKAda Quartz</i> avec la bibliothèque <i>XNAdaLib</i>	3
3.	Construire <i>GTK Quartz</i> pour la bibliothèque <i>XNAdaLib</i>	4
4.	Construire <i>GTKAda</i> (sans <i>OpenGL</i> )	11
5.	Construire <i>GLADE</i>	13
6.	Construire <i>Simple Components</i> et <i>AICWL</i>	16
7.	Construire <i>Template-Parser</i>	18
8.	Construire <i>Gate3</i>	18

## 1. Configuration

Télécharger le fichier suivant sur le bureau du Mac :

Configuration : macOS 11.3, Xcode 12.5, GNAT CE 2021.

Voir leur installation sur Blady :

- macOS avec Xcode : [blady.pagesperso-orange.fr/liens.html#macosx](http://blady.pagesperso-orange.fr/liens.html#macosx)
- GNAT : [blady.pagesperso-orange.fr/creations.html#gnatosxinstall](http://blady.pagesperso-orange.fr/creations.html#gnatosxinstall)

## 2. Installer GTKAda Quartz avec la bibliothèque XNAdaLib

Télécharger le fichier suivant sur le bureau du Mac :

xnadalib-ce-2021-quartz-x86\_64-apple-darwin20.5.0-bin.tgz,

depuis le site de Source Forge "[sourceforge.net/projects/gnuada/files/](http://sourceforge.net/projects/gnuada/files/)

[GNAT\\_GPL%20Mac%20OS%20X/2021-big-sur](http://GNAT_GPL%20Mac%20OS%20X/2021-big-sur)".

Lancer le Terminal dans un compte administrateur et taper les commandes suivantes :

```
% instbase=/usr/local # chemin d'installation obligatoire pour les bibliothèques dynamiques
% archive=$HOME/Downloads
% cd $instbase
% sudo tar xzf $archive/xnadalib-ce-2021-quartz-x86_64-apple-darwin20.5.0-bin.tgz
```

Sur les derniers macOS, les fichiers issus d'Internet sont marqués en quarantaine, ils faut donc supprimer cet attribut :

```
% sudo xattr -r -d com.apple.quarantine xnadalib-ce-2021
```

GTK, GTKAda, Glade, Gate3, Template Parser, Simple Components, AICWL s'installent à partir du dossier : *\$instbase/xnadalib-ce-2021*.

Pour une utilisation courante, saisir aussi les commandes suivantes :

```
% instxada=$instbase/xnadalib-ce-2021
% echo 'PATH=$instxada/bin:$PATH' >> ~/.profile
% echo 'PATH=$instxada/bin:$PATH' >> ~/.bashrc
% echo 'export MANPATH=$instxada/man:$instxada/share/man:$MANPATH' >> ~/.profile
% echo 'export MANPATH=$instxada/man:$instxada/share/man:$MANPATH' >> ~/.bashrc
% echo 'export GPR_PROJECT_PATH=$instxada/lib/gnat:$instxada/share/gpr:$GPR_PROJECT_PATH'
>> ~/.profile
% echo 'export GPR_PROJECT_PATH=$instxada/lib/gnat:$instxada/share/gpr:$GPR_PROJECT_PATH'
>> ~/.bashrc
% echo 'export XDG_DATA_DIRS=$instxada/share' >> ~/.profile
% echo 'export XDG_DATA_DIRS=$instxada/share' >> ~/.bashrc
```

Pour une utilisation temporaire, utiliser à chaque fois les commandes suivantes :

```
% instxada=$instbase/xnadalib-ce-2021
% PATH=$instxada/bin:$PATH
% export MANPATH=$instxada/man:$instxada/share/man:$MANPATH
```

```
% export GPR_PROJECT_PATH=$instxada/lib/gnat:$instxada/share/gpr:$GPR_PROJECT_PATH
% export XDG_DATA_DIRS=$instxada/share
```

Des exemples de programme Ada sont disponibles dans le dossier *\$instxada/share/examples/gtkada/testgtk*.

Une documentation au format HTML est disponible dans les dossiers *\$instxada/share/doc/gtkada* et *\$instxada/share/gtk-doc/html* :

```
% open $instxada/share/doc/gtkada/gtkada_rm/index.html
% open $instxada/share/doc/gtkada/gtkada_ug/index.html
```

Voir l'utilisation de *GTKAda* avec des exemples sur Blady :  
[blady.pagesperso-orange.fr/a\\_savoir.html#gtkada](http://blady.pagesperso-orange.fr/a_savoir.html#gtkada)

### 3. Construire GTK Quartz pour la bibliothèque XNAdaLib

*GTKAda* est la boîte à outil graphique en Ada basée sur *GTK+* pour construire des applications portables sur la plupart des plateformes.

Source [www.adacore.com/gtkada](http://www.adacore.com/gtkada).

Les descriptions des différents logiciels sont traduites des description inscrite sur leur site Web correspondant.

Attention : respecter chaque licence accompagnant les logiciels.

Récupérer les archives suivantes depuis le site Community d'AdaCore [www.adacore.com/download](http://www.adacore.com/download) à la page *More packages...*, sélectionner plateforme *x86 GNU Linux (64 bits)* et *2021* (même si *GtkAda* n'a pas été intégré pour *macOS / Darwin*, les sources pour *Linux* sont utilisables) puis déplier *GtkAda* -> *Sources* :

Liste des sources

<b>atk-2.15.1-22.0w-20210419-154DC-src.tar.gz</b>	793.4 KiB	Apr 20 2021
<b>cairo-1.14.6-22.0w-20210419-1560D-src.tar.gz</b>	41.7 MiB	Apr 20 2021
<b>fontconfig-2.13.1-22.0w-20210419-15ABF-src.tar.gz</b>	2 MiB	Apr 20 2021
<b>freetype-2.9.1.tar.gz</b>	2.4 MiB	Oct 24 2018
<b>gdk-pixbuf-2.30.8-22.0w-20210419-15A91-src.tar.gz</b>	2.4 MiB	Apr 20 2021
<b>gettext-0.19.6.tar.gz</b>	17.8 MiB	May 02 2019
<b>glib-2.44.1-22.0w-20210419-15680-src.tar.gz</b>	9.2 MiB	Apr 20 2021
<b>gobject-introspection-1.42.0-22.0w-20210419-1571A-src.tar.gz</b>	1.9 MiB	Apr 20 2021
<b>gtk+-3.14.15-22.0w-20210419-154FA-src.tar.gz</b>	30.1 MiB	Apr 20 2021
<b>harfbuzz-1.3.4-22.0w-20210419-154F6-src.tar.gz</b>	1.7 MiB	Apr 20 2021
<b>libffi-3.2.1-patched.tar.gz</b>	916.7 KiB	Sep 19 2018
<b>libjpeg-turbo-1.3.0.tar.gz</b>	1.3 MiB	Dec 10 2015
<b>libpng-1.2.44.tar.gz</b>	809.6 KiB	Dec 10 2015
<b>pango-1.36.8-22.0w-20210419-1546C-src.tar.gz</b>	1.7 MiB	Apr 20 2021
<b>pixman-0.33.0-22.0w-20210419-154E8-src.tar.gz</b>	1 MiB	Apr 20 2021
<b>pkg-config-0.29.1.tar.gz</b>	1.9 MiB	Feb 07 2017

Et aussi quelques manquants : *libtiff*, *autoconf*, *automake*, *libtool* qui seront téléchargés au moment de leur construction.

Et aussi *xnadalib-ce-2021-diff.zip* sur [blady.pagesperso-orange.fr/telechargements/gtkada/xnadalib-ce-2021-diff.zip](http://blady.pagesperso-orange.fr/telechargements/gtkada/xnadalib-ce-2021-diff.zip).

Saisir les commandes suivantes dans le Terminal :

```
% instbase=/usr/local # obligatoire pour les bibliothèques dynamiques
% version=ce-2021
% cd $instbase
% sudo mkdir src-$version
% sudo chown $USER src-$version
% xnadasrc=$instbase/src-$version
% sudo mkdir xnadalib-$version
% sudo chown $USER xnadalib-$version
% instxada=$instbase/xnadalib-$version
% archive=$HOME/Downloads
```

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

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

Saisir les commandes suivantes :

```
# Récupérer les correctifs apportés sur Blady :
% cd $archive
% unzip xnadalib-ce-2021-diff.zip
% export XML_CATALOG_FILES=$instxada/share/xml/catalog
% PATH=$instxada/bin:$PATH
```

### a) Construire *Pkg-Config*

*Pkg-config* est un utilitaire utilisé pour la gestion de bibliothèques en insérant les bonnes options de compilation.

(Dépendances : sans)

```
% cd $xnadasrc
% tar xzf $archive/pkg-config-0.29.1.tar.gz
% cd pkg-config-0.29.1
% ./configure --prefix=$instxada --with-internal-glib
% make -w
% make install
```

### b) Construire *Gettext* - *LibINTL*

Le but de cette bibliothèque est de donner une interface unique pour la gestion des messages avec des possibilités d'internationalisation.

(Dépendances : sans)

```
% cd $xnadasrc
% tar xzf $archive/gettext-0.19.6.tar.gz
% cd gettext-0.19.6
% ./configure --prefix=$instxada
% make -w
% make install
```

### c) Construire LibFFI

Le but de la bibliothèque Foreign Function Interface (FFI) est de fournir la possibilité d'interfacer des langages ayant différentes conventions de passages de paramètres.

(Dépendances : pkg-config)

```
% cd $xnadasrc
% tar xzf $archive/libffi-3.2.1-patched.tar.gz
% cd libffi-3.2.1
% ./configure --prefix=$instxada --disable-buildir
% make -w
% make install
```

### d) Construire Autoconf

Le but de cet utilitaire est de produire un script shell de configuration de construction de logiciels à partir de macros M4.

(Dépendances : sans)

Récupérer l'archive suivante à partir du site <ftp.gnu.org/gnu/autoconf> :

```
- autoconf-2.65.tar.xz      2009-11-21 09:51  1.1M
```

```
% cd $xnadasrc
% tar xzf $archive/autoconf-2.65.tar.xz
% cd autoconf-2.65
% ./configure --prefix=$instxada
% make -w
% make install
```

### e) Construire Automake

Le but de cet utilitaire est de produire un makefile portable.

(Dépendances : sans)

Récupérer l'archive suivante à partir du site <ftp.gnu.org/gnu/automake> :

```
- automake-1.15.1.tar.xz    2017-06-19 16:23  1.4M
```

```
% cd $xnadasrc
% tar xzf $archive/automake-1.15.1.tar.xz
% cd automake-1.15.1
% ./configure --prefix=$instxada
% make -w
% make install
```

### f) Construire Libtool

Le but de cet utilitaire est de produire des bibliothèques dynamiques portables.

(Dépendances : sans)

Récupérer l'archive suivante à partir du site <ftp.gnu.org/gnu/libtool> :

- libtool-2.4.2.tar.xz 2011-10-18 04:22 848K

```
% cd $xnadasrc
% tar xzf $archive/libtool-2.4.2.tar.xz
% cd libtool-2.4.2
% ./configure --prefix=$instxada
% make -w
% make install
```

### g) Construire GLib

GLib est la bibliothèque de bas niveau à la base des projets GTK+ et GNOME.  
(Dépendances : pkg-config, gettext - libintl, libffi)

```
% cd $xnadasrc
% tar xzf $archive/glib-2.44.1-22.0w-20210419-15680-src.tar.gz
% cd glib-2.44.1-22.0w-20210419-15680-src
% CPPFLAGS=-I$instxada/include LDFLAGS=-L$instxada/lib ./configure --prefix=$instxada --disable-gtk-
doc
% make -w
% make install
```

### h) Construire GObject-Introspection

Ce projet est d'abord de fournir une structure de description d'APIs d'une bibliothèque dans un langage codifié. Le format XML retenu est nommé GIR. Il n'est pas orienté sur un langage spécifique. Des utilitaires permettent de scanner les APIs ou de compiler les fichiers GIR.  
(Dépendances : pkg-config, libffi, glib)

```
% cd $xnadasrc
% tar xzf $archive/gobject-introspection-1.42.0-22.0w-20210419-1571A-src.tar.gz
% cd gobject-introspection-1.42.0-22.0w-20210419-1571A-src
% ./configure --prefix=$instxada
# Patch giscanner/utils.py, giscanner/utils.py, giscanner/sourcescanner.py sur Blady
% GI_SCANNER_INCLUDES=-I$instxada/include make -w
% make install
```

### i) Construire Atk

Accessibility Toolkit.

(Dépendances : pkg-config, glib)

```
% cd $xnadasrc
% tar xzf $archive/atk-2.15.1-22.0w-20210419-154DC-src.tar.gz
% cd atk-2.15.1-22.0w-20210419-154DC-src
% CPPFLAGS=-I$instxada/include LDFLAGS=-L$instxada/lib ./configure --prefix=$instxada
% make -w
% make install
```

### j) Construire Freetype

FreeType 2 est un logiciel de gestion de polices de caractères portable.

(Dépendances : pkg-config)

```
% cd $xnadasrc
% tar xzf $archive/freetype-2.9.1.tar.gz
```

```
% cd freetype-2.9.1
% ./configure --prefix=$instxada
% make -w
% make install
```

### **k) Construire Pixman**

Pixman est une bibliothèque qui fournit des fonctions bas niveau de manipulation de pixels.  
(Dépendances : pkg-config)

```
% cd $xnadasrc
% tar xzf $archive/pixman-0.33.0-22.0w-20210419-154E8-src.tar.gz
% cd pixman-0.33.0-22.0w-20210419-154E8-src
% ./configure --prefix=$instxada
# Patch test/utls-prng.c sur Blady
% make -w
% make install
```

### **l) Construire Fontconfig**

Bibliothèque de configuration de polices de caractères.  
(Dépendances : pkg-config, freetype)

```
% cd $xnadasrc
% tar xzf $archive/fontconfig-2.13.1-22.0w-20210419-15ABF-src.tar.gz
% cd fontconfig-2.13.1-22.0w-20210419-15ABF-src
% ./configure --prefix=$instxada
% make -w
% make install
```

### **m) Construire LibPNG**

Bibliothèque officielle du format PNG.  
(Dépendances : pkg-config)

```
% cd $xnadasrc
% tar xzf $archive/libpng-1.2.44.tar.gz
% cd libpng-1.2.44
% ./configure --prefix=$instxada
% make -w
% make install
```

### **n) Construire Cairo**

Cairo est une bibliothèque graphique 2D pour plusieurs dispositifs d'affichage comme X Window System, win32, Quartz ainsi que les formats de fichier PDF, PostScript, and SVG (OpenGL n'est pas actif).

(Dépendances : pkg-config, pixman, libpng, freetype, fontconfig, glib)

```
% cd $xnadasrc
% tar xzf $archive/cairo-1.14.6-22.0w-20210419-1560D-src.tar.gz
% cd cairo-1.14.6-22.0w-20210419-1560D-src
% ./configure --prefix=$instxada
# Patch src/cairo-quartz-surface.c sur Blady
% make -w
% make install
```



### **o) Construire Harfbuzz**

Harfbuzz est un moteur de dessin de textes OpenType.

(Dépendances : pkg-config, glib, cairo, freetype)

```
% cd $xnadasrc
% tar xzf $archive/harfbuzz-1.3.4-22.0w-20210419-154F6-src.tar.gz
% cd harfbuzz-1.3.4-22.0w-20210419-154F6-src
% ./configure --prefix=$instxada
% make -w
% make install
```

### **p) Construire Pango**

Pango est une bibliothèque pour le rendu et l'affichage de textes avec des possibilités d'internationalisation.

(Dépendances : pkg-config, fontconfig, cairo)

```
% cd $xnadasrc
% tar xzf $archive/pango-1.36.8-22.0w-20210419-1546C-src.tar.gz
% cd pango-1.36.8-22.0w-20210419-1546C-src
% ./configure --prefix=$instxada
% make -w
% make install
```

### **q) Construire LibJPEG**

LibJPEG-Turbo est une bibliothèque compatible avec libjpeg v8 (IJG) qui utilise les instructions d'accélération des processeurs pour les opérations de compression et de décompression d'images JPEG.

IJG est un groupe informel qui écrit et distribue une bibliothèque libre pour les images JPEG.

(Dépendances : NASM)

```
% cd $xnadasrc
% tar xzf $archive/libjpeg-turbo-1.3.0.tar.gz
% cd libjpeg-turbo-1.3.0
# Utilisation de NASM de MacPorts
% ln -s /opt/local/bin/nasm $instxada/bin
% ./configure --prefix=$instxada --without-simd
% make -w
% make install
```

### **r) Construire LibTiff**

LibTiff est une bibliothèque pour la manipulation des images au format TIFF (Tag Image File Format, OpenGL n'est pas actif).

(Dépendances : pkg-config)

Récupérer l'archive suivante à partir du site [download.osgeo.org/libtiff](http://download.osgeo.org/libtiff) :

- tiff-4.0.5.zip 2M 2015-Aug-31 00:03

```
% cd $xnadasrc
% tar xzf $archive/tiff-4.0.5.tar.gz
% cd tiff-4.0.5
```

```
% ./configure --prefix=$instxada
% make -w
% make install
```

### s) Construire GDK-PixBuf

GDK-PixBuf est une bibliothèque pour le tracé d'image dans une mémoire graphique tampon.  
(Dépendances : pkg-config, glib, libjpeg, libtiff, libpng)

```
% cd $xnadasrc
% tar xzf $archive/gdk-pixbuf-2.30.8-22.0w-20210419-15A91-src.tar.gz
% cd gdk-pixbuf-2.30.8-22.0w-20210419-15A91-src
% CPPFLAGS=-I$instxada/include LDFLAGS=-L$instxada/lib ./configure --prefix=$instxada
% make -w
% make install
```

### t) Construire GTK

Gtk est une bibliothèque graphique en C pour X-Window et Win32. Elle fut développée initialement pour Gimp.

(Dépendances : pkg-config, glib, libjpeg, libtiff, gdk-pixbuf, pango, cairo, atk)

```
% cd $xnadasrc
% tar xzf $archive/gtk+-3.14.15-22.0w-20210419-154FA-src.tar.gz
% cd gtk+-3.14.15-22.0w-20210419-154FA-src
% ./configure --prefix=$instxada --enable-quartz-backend --enable-quartz-relocation --disable-cups
% make -w
% make install
```

Des programme de test et démo donnent plein de situations d'emploi des objets GTK avec leur description, le code source C correspondant et une démonstration du résultat :

```
% export XDG_DATA_DIRS=$instxada/share
% cd $instxada/bin
% ./gtk3-demo
```

## 4. Construire GTKAda (sans OpenGL)

GTKAda est la boîte à outil graphique en Ada basée sur GTK pour construire des applications portables sur la plupart des plateformes.

Site Web : [www.adacore.com/gtkada](http://www.adacore.com/gtkada)

La version installée est CE 2021gtk.

(Dépendances : pkg-config, gtk)

Saisir les commandes suivantes dans le Terminal :

```
% instbase=/usr/local # obligatoire pour les bibliothèques dynamiques
% version=ce-2021
% xnadasrc=$instbase/src-$version
% instxada=$instbase/xnadalib-$version
% archive=$HOME/Downloads
```

Avant de démarrer la compilation, GNAT et le dossier d'installation doivent être activés, ajouter leur emplacement comme par exemple (si pas déjà fait) :

```
# Activation du compilateur Ada GNAT
% PATH=/opt/gnat-ce-2021/bin:$PATH
# Activation de la bibliothèque GTK-OSX
% PATH=$instxada/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% tar xzf $archive/gtkada-2021-20210518-1A011-src.tar.gz
% cd gtkada-2021-20210518-1A011-src
% ./configure --prefix=$instxada
# Patch src/generated/gtk-list_store.adb sur Blady
# On applique les corrections de Blady pour le programme de test
% cd $archive
% unzip gtkada-tests.zip
% cd gtkada-tests
% for i in * ; do cp -p $i $xnadasrc/gtkada-2021-20210518-1A011-src/testgtk; done
% make -w
% make install
% cp -p po/build_skeleton.pl $instxada/bin
# Utilisation de gnatdoc de CE 2019
% PATH=/usr/local/adacore/2019/bin:$PATH
# Utilisation de Sphinx de MacPorts
% PATH=/opt/local/bin:$PATH make docs
% make install
```

La bibliothèque GTKAda s'est installée dans le dossier \$instxada.

Pour une utilisation courante, nous positionnons les variables d'environnement PATH pour une utilisation en ligne de commande et GPR\_PROJECT\_PATH pour une utilisation avec un projet GPS :

```
% echo 'PATH=$instxada/bin:$PATH' >> ~/.profile
% echo 'PATH=$instxada/bin:$PATH' >> ~/.bashrc
% echo 'export MANPATH=$instxada/man:$MANPATH' >> ~/.profile
```

```
% echo 'export MANPATH=$instxada/man:$MANPATH' >> ~/.bashrc
% echo 'export MANPATH=$instxada/share/man:$MANPATH' >> ~/.profile
% echo 'export MANPATH=$instxada/share/man:$MANPATH' >> ~/.bashrc
% echo 'export GPR_PROJECT_PATH=$instxada/lib/gnat:$instxada/share/gpr:$GPR_PROJECT_PATH'
>> ~/.profile
% echo 'export GPR_PROJECT_PATH=$instxada/lib/gnat:$instxada/share/gpr:$GPR_PROJECT_PATH'
>> ~/.bashrc
```

Pour une utilisation temporaire, utiliser à chaque fois les commandes suivantes :

```
% PATH=$instxada/bin:$PATH
% export MANPATH=$instxada/man:$MANPATH
% export MANPATH=$instxada/share/man:$MANPATH
% export GPR_PROJECT_PATH=$instxada/lib/gnat:$instxada/share/gpr:$GPR_PROJECT_PATH
```

Une démo tout en Ada est présente dans le dossier "\$instxada/share/examples/gtkada" :

```
% export XDG_DATA_DIRS=$instxada/share
% cd $instxada/share/examples/gtkada/testgtk
% ./testgtk
```

Une documentation au format PDF et HTML sous forme d'un manuel utilisateur et d'un manuel de référence est disponible dans le dossier "\$instxada/share/doc/gtkada".

```
% open $instxada/share/doc/gtkada/gtkada_ug/GtkAda.pdf
% open $instxada/share/doc/gtkada/gtkada_ug/index.html
% open $instxada/share/doc/gtkada/gtkada_rm/index.html
```

Voir l'utilisation de *GTKAda* avec des exemples sur Blady :  
[blady.pagesperso-orange.fr/a\\_savoir.html#gtkada](http://blady.pagesperso-orange.fr/a_savoir.html#gtkada)

## 5. Construire GLADE

Glade est un outil graphique de développement d'interfaces utilisateurs pour la bibliothèque GTK. Les fichiers XML produits par Glade peuvent être utilisés par de nombreux langage de programmation comme C, C++, C#, Vala, Java, Perl, Python et ... Ada ;-).

Site web : [glade.gnome.org](http://glade.gnome.org).

La version installée est 3.16.1.

Saisir les commandes suivantes dans le Terminal :

```
% instbase=/usr/local # obligatoire pour les bibliothèques dynamiques
% version=ce-2021
% xnadasrc=$instbase/src-$version
% instxada=$instbase/xnadalib-$version
% archive=$HOME/Downloads
```

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

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

Saisir les commandes suivantes tout en étant connecté à Internet :

```
% export XML_CATALOG_FILES=$instxada/share/xml/catalog
% PATH=$instxada/bin:$PATH
```

### a) Construire Hicolor-Icon-Theme

Thème par défaut pour la définition des icônes.

(Dépendances : pkg-config)

```
% cd $xnadasrc
% tar xzf $archive/hicolor-icon-theme-0.12.tar.gz
% cd hicolor-icon-theme-0.12
% ./configure --prefix=$instxada
% make -w
% make install
```

### b) Construire IntlTool

Ensemble d'outils facilitants la traduction de fichiers de différents format.

(Dépendances : pkg-config)

```
% cd $xnadasrc
% tar xzf $archive/intltool-0.41.1.tar.gz
% cd intltool-0.41.1
% ./configure --prefix=$instxada
% make -w
% make install
```

### c) Construire adwaita-icon-theme

Thème pour la définition des icônes notamment de l'interface graphique GNOME.

(Dépendances : pkg-config)

```
% cd $xnadasrc
% tar xzf $archive/adwaita-icon-theme-3.14.1.tar.gz
% cd adwaita-icon-theme-3.14.1
% ./configure --prefix=$instxada
% make -w
% make install
```

#### **d) Construire LibXML2**

Bibliothèque de lecture et de décodage des fichiers XML  
(Dépendances : pkg-config)

```
% cd $xnadasrc
% tar xzf $archive/libxml2-2.9.4.tar.gz
% cd libxml2-2.9.4
% ./configure --prefix=$instxada --libdir=$instxada/lib --with-python --with-python-install-dir=$instxada/lib/
python3.8/site-packages
% make -w
% make install
```

#### **e) Construire LibCroco**

Bibliothèque de lecture et de décodage des fichiers CSS2 et un moteur expérimental de rendu XML / CSS.  
(Dépendances : pkg-config, glib, libxml)

```
% cd $xnadasrc
% tar xzf $archive/libcroco-0.6.8.tar.gz
% cd libcroco-0.6.8
% ./configure --prefix=$instxada --disable-Bsymbolic
% make -w
% make install
```

#### **f) Construire LibRSVG**

Bibliothèque de rendu SVG associée au projet GNOME.  
(Dépendances : pkg-config, gdk-pixbuf, glib, libxml, pango, cairo, fontconfig, freetype)

```
% cd $xnadasrc
% tar xzf $archive/librsvg-2.40.6.tar.gz
% cd librsvg-2.40.6
% ./configure --prefix=$instxada --disable-Bsymbolic
% make -w
% make install
```

#### **g) Construire GTKMacIntegration**

Donne la possibilité d'intégrer une application GTK avec le bureau du Mac..  
(Dépendances : pkg-config)

Récupérer l'archive suivante à partir du site [github.com/jralls/gtk-mac-integration/releases/tag/gtk-mac-integration-2.0.8](https://github.com/jralls/gtk-mac-integration/releases/tag/gtk-mac-integration-2.0.8)  
- gtk-mac-integration-2.0.8.tar.xz

```
% cd $xnadasrc
% tar xzf $archive/gtk-mac-integration-2.0.8.tar.xz
```

```
% cd gtk-mac-integration-2.0.8
% ./configure --prefix=$instxada --disable-python
# Patch src/test-integration.c sur Blady
% make -w
% make install
```

#### **h) Construire Glade**

Glade est un outil graphique de développement d'interfaces utilisateurs pour la bibliothèque GTK. Les fichiers XML produits par Glade peuvent être utilisés par de nombreux langage de programmation comme C, C++, C#, Java, Perl, Python et ... Ada ;-)

(Dépendances : glib, gtk+, libxml)

```
% cd $xnadasrc
% cd glade-3.16.1-22.0w-20210419-15735-src
% ./configure --prefix=$instxada
# Patch src/glade-window.c sur Blady
% make -w
% make install
```

Ne pas oublier de configurer LANG (pour l'affichage français) et XDG\_DATA\_DIRS avant de lancer Glade :

```
% export XDG_DATA_DIRS=$instxada/share
% export LANG=fr_FR.UTF-8
% $instxada/bin/glade
```

Voir sur Blady pour les premiers pas avec un exemple :  
[blady.pagesperso-orange.fr/a\\_savoir.html#gtkada](http://blady.pagesperso-orange.fr/a_savoir.html#gtkada)

## 6. Construire Simple Components et AICWL

Simple Components et AICWL proposés par Dmitry Kazakov contiennent les composants :

- Simple components : graphes, ensembles, piles, vecteurs, analyseurs d'expressions, primitives de synchronisation, nombre pseudo-aléatoires...
- Strings edit : mise à l'échelle des axes, nombre romains, entiers et réels, codage UTF-8 et Unicode, recherche avec jokers...
- Tables : container de données avec recherche par chaînes de caractères,
- AICWL (Ada industrial control widget library) : collection de widgets de visualisation type compteur de vitesse et vue mètre, d'horloges, d'oscillogramme, éditeur de widget...
- GtkAda contributions : multi-tâche, vue arborescente, navigation de fichiers, image en code source Ada, fichier de ressources graphiques, modèle de couleur HSL, des boutons, exécution de processus asynchrones...

Site web : [www.dmitry-kazakov.de](http://www.dmitry-kazakov.de).

### a) Construire Simple Components

Site web : [www.dmitry-kazakov.de/ada/components.htm](http://www.dmitry-kazakov.de/ada/components.htm).

Version installée : 4.50.

Récupérer l'archive suivante :

[sourceforge.net/projects/simplecomponentsforada/files/releases/components\\_4\\_50.tgz/download](http://sourceforge.net/projects/simplecomponentsforada/files/releases/components_4_50.tgz/download)

Avant de démarrer la compilation, GNAT et le dossier d'installation de XNAdaLib doivent être activés, ajouter leur emplacement comme par exemple (si pas déjà fait) :

```
% instbase=/usr/local # obligatoire pour les bibliothèques dynamiques
% version=ce-2021
% xnadasrc=$instbase/src-$version
% instxada=$instbase/xnadalib-$version
% archive=$HOME/Downloads
% PATH=$instxada/bin:$PATH
% PATH=/opt/gnat-ce-2021/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% mkdir components_4_50
% cd components_4_50
% tar xzf $archive/components_4_50.tgz
# Récupérer les projets GPR sur Blady
% unzip $archive/xnadalib-ce-2021-diff/components_gpr
% gprbuild -p -P lib_components.gpr -XSC_OS=OSX
% gprinstall -f -p --prefix=$instxada -P lib_components.gpr -XSC_OS=OSX
```

Une documentation au format HTML est disponible dans le dossier \$instxada/share/doc/components :

```
% open $instxada/share/doc/components/strings_edit.htm
% open $instxada/share/doc/components/tables.htm
% open $instxada/share/doc/components/components_rm/index.html
```



Des exemples de programme sont disponibles dans le dossier `$instxada/share/examples/components`.

## b) Construire AICWL

Site web : [www.dmitry-kazakov.de/ada/aicwl.htm](http://www.dmitry-kazakov.de/ada/aicwl.htm).

Version installée : 3.24.

Récupérer l'archive suivante sur le bureau :

[sourceforge.net/projects/aicwl/files/releases/aicwl\\_3\\_24.tgz/download](https://sourceforge.net/projects/aicwl/files/releases/aicwl_3_24.tgz/download)

Avant de démarrer la compilation, GNAT et le dossier d'installation de XNAdaLib doivent être activés, ajouter leur emplacement comme par exemple (si pas déjà fait) :

```
% instbase=/usr/local # obligatoire pour les bibliothèques dynamiques
% version=ce-2021
% xnadasrc=$instbase/src-$version
% instxada=$instbase/xnadalib-$version
% archive=$HOME/Downloads
% PATH=$instxada/bin:$PATH
% export GPR_PROJECT_PATH=$instxada/lib/gnat:$instxada/share/gpr
% PATH=/opt/gnat-ce-2021/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% mkdir aicwl_3_24
% cd aicwl_3_24
% tar xzf $archive/aicwl_3_24.tgz
# Récupérer les projets GPR sur Blady
% unzip $archive/xnadalib-ce-2021-diff/aicwl_gpr.zip
% gprbuild -p -P lib_aicwl.gpr -XSC_OS=OSX
% gprinstall -f -p --prefix=$instxada -P lib_aicwl.gpr -XSC_OS=OSX
% gprbuild -p -P xpm2gtkada/build_xpm2gtkada.gpr -XSC_OS=OSX
% gprinstall -f -p --prefix=$instxada -P xpm2gtkada/build_xpm2gtkada.gpr -XSC_OS=OSX
```

Une documentation au format HTML est disponible dans le dossier `$instxada/share/doc/aicwl` :

```
% open $instxada/share/doc/aicwl/aicwl.htm
% open $instxada/share/doc/aicwl/gtkada_contributions.htm
% open $instxada/share/doc/aicwl/aicwl_rm/index.html
```

Des exemples de programme sont disponibles dans le dossier `$instxada/share/examples/aicwl`.

Ne pas oublier de configurer `XDG_DATA_DIRS` avant de lancer les exemples :

```
% cd $instxada/share/examples/aicwl
% gprbuild -p -P build_examples.gpr -XSC_OS=OSX
% export XDG_DATA_DIRS=$instxada/share
% ./bin/oscilloscope_plotter
```

...

## 7. Construire Template-Parser

Il s'agit d'un composant qui remplace des zones de textes balisées dans des modèles.

Site Web : [github.com/AdaCore/templates-parser](https://github.com/AdaCore/templates-parser)

La version installée est 21.2.

Dépendance : XMLAda (optionnel).

Avant de démarrer la compilation, GNAT et le dossier d'installation de XNAdaLib doivent être activés, ajouter leur emplacement comme par exemple :

```
% instbase=/usr/local # obligatoire pour les bibliothèques dynamiques
% version=ce-2021
% xnadasrc=$instbase/src-$version
% instxada=$instbase/xnadalib-$version
% archive=$HOME/Downloads
% PATH=/opt/gnat-ce-2021/bin:$PATH

% cd $xnadasrc
% git clone https://github.com/AdaCore/templates-parser.git -b 21.2
% cd templates-parser
% make prefix=$instxada setup
% make
% make install
```

## 8. Construire Gate3

Gate3 est un utilitaire qui produit du code Ada à partir d'un fichier Glade.

Gate3 a été développé par Francois Fabien sous licence MIT.

Site web : [sourceforge.net/projects/lorenz](https://sourceforge.net/projects/lorenz).

La version installée est 0.5c.

Avant de démarrer la compilation, GNAT et le dossier d'installation de XNAdaLib doivent être activés, ajouter leur emplacement comme par exemple (si pas déjà fait) :

```
% instbase=/usr/local # obligatoire pour les bibliothèques dynamiques
% version=ce-2021
% xnadasrc=$instbase/src-$version
% instxada=$instbase/xnadalib-$version
% archive=$HOME/Downloads
% PATH=$instxada/bin:$PATH
% export GPR_PROJECT_PATH=$instxada/lib/gnat:$instxada/share/gpr
% PATH=/opt/gnat-ce-2021/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd /usr/local/src-2020
% git clone https://github.com/Blady-Com/gate3.git
% cd gate3
% make
% make PREFIX=$instxada install
```

```
% cd $xnadasrc
% env
% git clone https://github.com/Blady-Com/gate3.git
% cd gate3
% make
% make PREFIX=$instxada install
```

Des exemples sont construits avec :

```
% make editor
% make calculator
% make lady
% make lorenz
```

Leur exécution :

```
% ./bin/editor
% ./bin/calculator
% ./bin/lady
% ./bin/lorenz
```

Les sources Ada sont construits par Gate3 à partir d'un fichier Glade avec gate3.sh. Voir le tutoriel Factoriel sur Blady :

[blady.pagesperso-orange.fr/a\\_savoir.html#gtkada](http://blady.pagesperso-orange.fr/a_savoir.html#gtkada)

Une documentation au format texte est disponible dans le dossier \$instxada/share/gate3/doc. Un tutoriel mettant en oeuvre l'application factorielle est présent dans le dossier \$instxada/share/gate3/tutorial.

Les fichiers modèles pour la génération des sources Ada dans le dossier \$instxada/share/gate3/tmpl peuvent être modifiés en incluant notamment la licence dans les modèles des paquetages spécifications et corps :

- gate3\_license.txt : licence MIT par défaut, à changer à votre convenance
- gate3\_header.tmpl : en-tête de la procédure principale, inclut gate3\_license.txt
- gate3\_main.tmpl : modèle de la procédure principale
- gate3\_spec.tmpl : modèle du paquetage spécification des callbacks
- gate3\_body.tmpl : modèle du paquetage corps des callbacks

Pascal Pignard, avril-septembre 2011, août 2014, juillet 2015, août-septembre 2016, octobre 2017, septembre 2018, octobre 2020, août 2021.

<http://blady.pagesperso-orange.fr>