

Installer ou construire GTKAda Quartz et autres composants en Ada sur macOS avec 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.

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/docs/installations/macos.

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 : Gate3, Florist, Simple Components, AICWL, AdaCurses, Zanyblue, PragmARC, UXStrings, Gnoga, SparForte et HAC.

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 GTKAda Quartz avec la bibliothèque XNAdaLib	3
3.	Construire GTK Quartz pour la bibliothèque XNAdaLib	4
4.	Construire GTKAda	8
5.	Construire GLADE	9
6.	Construire Simple Components et AICWL	10
7.	Construire Templates-Parser	13
8.	Construire Gate3	13
9.	Construire Florist	14
10.	Construire AdaCurses	15
11.	Construire Zanyblue	16
12.	Construire PragmARC	17
13.	Construire UXStrings	17
14.	Construire Gnoga	18
15.	Construire SparForte	19
16.	Construire HAC	20

1. Configuration

Configuration : macOS 12.6, Xcode 13.4.1, GNAT FSF 12.1.

Voir leur installation sur Blady :

- macOS et Xcode : blady.pagesperso-orange.fr/liens.html#macosx
- GNAT : blady.pagesperso-orange.fr/creations.html#gnaatosxinstall

2. Installer GTKAda Quartz avec la bibliothèque XNAdaLib

Télécharger le fichier suivant sur le bureau du Mac :

`xnadalib-2022-quartz-x86_64-apple-darwin21.6.0.pkg`,

depuis le site de Source Forge "sourceforge.net/projects/gnuada/files/GNAT_GPL%20Mac%20OS%20X/2022-monterey".

Le fichier n'est pas signé alors ne doublez-cliquer pas dessus. Faites plutôt un clic-droit et cliquez sur *Ouvrir*. Accepter l'avertissement, vous êtes ensuite guidés dans l'installation.

GTK, GTKAda, Glade, Gate3, Florist, Simple Components, AICWL, AdaCurses, Zanyblue, PragmARC, UXStrings, Gnoga, SparForte, HAC s'installent à partir du dossier : `/usr/local/xnadalib-2022` (chemin d'installation obligatoire pour le bon fonctionnement des bibliothèques dynamiques macOS).

Pour une utilisation courante, saisir aussi les commandes suivantes :

```
% xnadabase=/usr/local # chemin d'installation obligatoire pour les bibliothèques dynamiques
% cd $xnadabase
% xnadainst=$xnadabase/xnadalib-2022
% echo 'PATH=$xnadainst/bin:$PATH' >> ~/.zshenv
% echo 'export MANPATH=$xnadainst/man:$xnadainst/share/man:$MANPATH' >> ~/.zshenv
% echo 'export GPR_PROJECT_PATH=$xnadainst/share/gpr:$GPR_PROJECT_PATH' >>
~/.zshenv
% echo 'export XDG_DATA_DIRS=$xnadainst/share' >> ~/.zshenv
```

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

```
% xnadabase=/usr/local # chemin d'installation obligatoire pour les bibliothèques dynamiques
% cd $xnadabase
% PATH=$xnadainst/bin:$PATH
% export MANPATH=$xnadainst/man:$xnadainst/share/man:$MANPATH
% export GPR_PROJECT_PATH=$xnadainst/share/gpr:$GPR_PROJECT_PATH
% export XDG_DATA_DIRS=$xnadainst/share
```

Des exemples de programme GTKAda sont disponibles dans le dossier `$xnadainst/share/examples/gtkada`.

Une documentation de GTKAda au format HTML est disponible dans les dossiers `$xnadainst/share/doc/gtkada` :

```
% open $xnadainst/share/doc/gtkada/gtkada_rm/index.html
% open $xnadainst/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
ainsi que l'utilisation de Gnoga :
blady.pagesperso-orange.fr/a_savoir.html#gnoga

3. Construire GTK Quartz pour la bibliothèque XNAdaLib

GTK est une bibliothèque graphique en C pour X-Window et Win32. Elle fut développée initialement pour Gimp. Nous allons construire la version comportant le rendu natif macOS directement avec Quartz.

Site web : www.gtk.org.

La version installée est 3.24.33.

Récupérer dans le dossier *Téléchargements* le script d'installation *gtk-osx-setup.sh* à l'adresse :

gitlab.gnome.org/GNOME/gtk-osx/raw/master/gtk-osx-setup.sh

Source wiki.gnome.org/Projects/GTK/OSX/Building.

Noter la date du téléchargement ou la date et le SHA du dernier commit sur :

gitlab.gnome.org/GNOME/gtk-osx/-/commits/master/gtk-osx-setup.sh.

Si vous avez un clone du dépôt vous pouvez apposer un TAG sur le dernier commit.

Et aussi *xnadalib-2022-diff.tgz* sur blady.pagesperso-orange.fr/telechargements/gtkada/xnadalib-2022-diff.tgz.

Saisir les commandes suivantes dans le Terminal d'une session administrateur tout en étant connecté à Internet :

```
% cd /usr/local # obligatoire pour les bibliothèques dynamiques
% xnadabase=$PWD
% version=2022
% cd $xnadabase
% sudo mkdir src-$version
% sudo chown $USER src-$version
% xnadasrc=$xnadabase/src-$version
% sudo mkdir xnadalib-$version
% sudo chown $USER xnadalib-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
```

Saisir les commandes suivantes dans le Terminal d'une session administrateur tout en étant connecté à Internet :

```
% export DEVROOT=$xnadasrc
% export PIP_CONFIG_DIR=$xnadasrc/config/pip
% export XDG_CONFIG_HOME=$xnadasrc/config
% export XDG_CACHE_HOME=$xnadasrc/cache
% cd $xnadasrc
% sh $xnarchive/gtk-osx-setup.sh
```

...
PATH does not contain .../src-2022/.new_local/bin. You probably want to fix that.

...
Cette commande installe *jhbuild* dans le dossier *Source* et crée le dossier *.new_local/bin* avec les utilitaires requis pour la suite. Il installe aussi les scripts d'installation *config/jhbuildrc* et *config/jhbuildrc-custom* et copie les modules *gtk-osx* courants dans *Source/jhbuild/modulesets*. (Si ces derniers fichiers sont déjà présents, je recommande de les supprimer avant de lancer la commande ci-dessus.)

Comme suggérer nous allons ajouter l'accès demandé dans notre environnement :

```
% echo 'PATH=$xnadasrc/.new_local/bin:$PATH' >> ~/.zshenv
```

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

```
% PATH=$xnadasrc/.new_local/bin:$PATH
```

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 :

```
# Appliquer les correctifs apportés sur Blady :  
% tar xzf $xnarchive/xnadalib-2022-diff.tgz  
% cd $xnadasrc/config  
% patch -p0 < $xnarchive/xnadalib-2022-diff/jhbuildrc-custom.diff  
% cd $xnadasrc
```

...
Lors de la première utilisation de Java, une fenêtre système surgit pour donner à Java l'accès au dossier Documents, cliquez sur OK

```
% jhbuild bootstrap-gtk-osx
```

```
...  
% jhbuild build meta-gtk-osx-bootstrap
```

```
...  
% jhbuild build meta-gtk-osx-gtk3
```

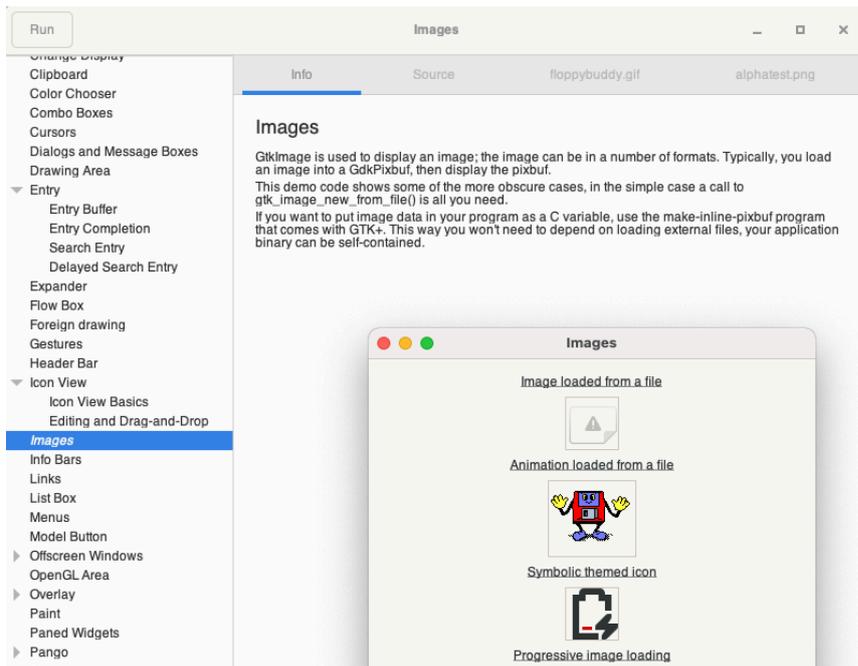
```
...
```

Liste des modules installés :

- adwaita-icon-theme-42.0
- atk-2.36.0
- autoconf-2.71
- autoconf-archive-2021.02.19
- automake-1.16.3
- bison-3.7.6
- cairo-1.17.6
- cmake-3.20.0
- flex-2.6.4
- fontconfig-2.13.1
- freetype-2.11.1
- fri bidi-1.0.11
- gdk-pixbuf-2.42.8
- gettext-0.21
- glib-2.72.0
- gobject-introspection-1.72.0
- gtk-doc-1.33.2
- gtk-mac-integration-3.0.1
- gtk-osx-docbook-1.3
- gtk+-3.24.33
- harfbuzz-4.1.0
- hicolor-icon-theme-0.17
- icu4c-68_2-src
- intltool-0.51.0
- itstool-2.0.6
- jpegsrc.v9e
- libepoxy-1.5.4
- libffi-3.4.3
- libpng-1.6.37
- librsvg-2.54.0
- libtool-2.4.6
- libxml2-2.9.12
- libxslt-1.1.34
- m4-1.4.18
- make-4.3
- pango-1.50.6
- pcre-8.45
- pixman-0.40.0
- pkg-config-0.29.2
- tiff-4.2.0
- util-macros-1.19.3
- xz-5.2.5
- zlib-1.2.13

Et surtout, 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 avec la commande :

```
% XDG_DATA_DIRS=$xnadainst/share $xnadainst/bin/gtk3-demo
```



4. Construire GTKAda

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

La version installée est 22.2.

Dépendances : GTK, ATK, Cairo, GDK-Pixbuf, Glib, Pango, Pixman, PNG.

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

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
# Activation de la bibliothèque GTK-OSX avec pkg-config
% PATH=$xnadainst/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% git clone https://github.com/Blady-Com/gtkada.git -b xnadalib-2022
% cd gtkada
% ./configure --prefix=$xnadainst --enable-build=Debug
% make -w GPRBUILD_OPTIONS=-gnatwn
# Utilisation de GNATDoc de GNAT CE 2019
# Utilisation de Sphinx et LaTeX avec MacPorts
% PATH=/usr/local/adacore/2019/bin:/opt/local/bin:$PATH make docs
% make install prefix=$xnadainst PRJDIR=share/gpr
% cp -p po/build_skeleton.pl $xnadainst/bin
```

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

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=$xnadainst/bin:$PATH' >> ~/.zshenv
% echo 'export MANPATH=$xnadainst/man:$MANPATH' >> ~/.zshenv
% echo 'export MANPATH=$xnadainst/share/man:$MANPATH' >> ~/.zshenv
% echo 'export GPR_PROJECT_PATH=$xnadainst/share/gpr:$GPR_PROJECT_PATH' >>
~/.zshenv
```

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

```
% PATH=$xnadainst/bin:$PATH
% export MANPATH=$xnadainst/man:$MANPATH
% export MANPATH=$xnadainst/share/man:$MANPATH
% export GPR_PROJECT_PATH=$xnadainst/share/gpr:$GPR_PROJECT_PATH
```

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

```
% cd $xnadainst/share/examples/gtkada/testgtk
% export XDG_DATA_DIRS=$xnadainst/share
% export GDK_PIXBUF_MODULE_FILE=$xnadainst/lib/gdk-pixbuf-2.0/2.10.0/loaders.cache
% export DYLD_LIBRARY_PATH=$xnadainst/lib/gdk-pixbuf-2.0/2.10.0/loaders
% ./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 `$xnadainst/share/doc/gtkada`.

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

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.

La version installée est 3.40.0.

Dépendances : GTK, itstool.

Saisir les commandes suivantes dans le Terminal :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% PATH=$xnadasrc/.new_local/bin:$PATH
% export PIP_CONFIG_DIR=$xnadasrc/config/pip
% export XDG_CACHE_HOME=$xnadasrc/cache
% export XDG_CONFIG_HOME=$xnadasrc/config
```

```
% jhbuild build glade
```

Ne pas oublier de configurer `LANG` (pour l'affichage français) et `XDG_DATA_DIRS` avant de lancer Glade :

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

Voir sur Blady pour les premiers pas avec un exemple :
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.

a) Construire GTKSourceView

Il s'agit d'un widget qui étend GtkTextView avec les fonctionnalités d'un éditeur de code source comme par exemple la coloration syntaxique.

Site web : wiki.gnome.org/Projects/GtkSourceView.

La version installée est 4.8.1.

Dépendances : GTK.

Saisir les commandes suivantes dans le Terminal :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% PATH=$xnadasrc/.new_local/bin:$PATH
% export PIP_CONFIG_DIR=$xnadasrc/config/pip
% export XDG_CONFIG_HOME=$xnadasrc/config
% export XDG_CACHE_HOME=$xnadasrc/cache
```

```
% jhbuild build gtksourceview3
```

```
...
```

b) Construire GNUTLS

Il s'agit d'une bibliothèque qui contient des API pour les protocoles SSL, TLS et DTLS.

Site web : www.gnutls.org.

La version installée est 3.7.7.

Dépendances : libnettle, libtasn1, zlib, p11-kit, libgcrypt.

Saisir la commande suivante dans le Terminal :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% PATH=$xnadasrc/.new_local/bin:$PATH
```

```
% export PIP_CONFIG_DIR=$xnadasrc/config/pip
% export XDG_CONFIG_HOME=$xnadasrc/config
% export XDG_CACHE_HOME=$xnadasrc/cache
```

```
% jhbuild build gnutls
```

```
...
```

c) Construire Simple Components

Site web : www.dmitry-kazakov.de/ada/components.htm.

Version installée : 4.64.

Dépendances : GNUTLS.

Récupérer l'archive suivante sur le bureau :

sourceforge.net/projects/simplecomponentsforada/files/releases/components_4_64.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) :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
# Activation de la bibliothèque GTK-OSX avec pkg-config
% PATH=$xnadainst/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% mkdir components_4_64
% cd components_4_64
% tar xzf $xnarchive/components_4_64.tgz
# Récupérer les projets GPR apportés sur Blady
% unzip $xnarchive/xnadalib-2022-diff/components_gpr.zip
% gprbuild -p -P lib_components.gpr -XSC_OS=OSX
# Utilisation de gnatdoc de GNAT CE 2019
# Certains fichiers sources provoquant une erreur avec gnatdoc sont ignorés
% /usr/local/adacore/2019/bin/gnatdoc --enable-build -P lib_components.gpr -XSC_OS=OSX
% gprinstall -f -p --prefix=$xnadainst -P lib_components.gpr -XSC_OS=OSX
```

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

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

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

d) Construire AICWL

Site web : www.dmitry-kazakov.de/ada/aicwl.htm.

Version installée : 3.25.

Dépendances : GTKAda, GTKSourceView.

Récupérer l'archive suivante sur le bureau :

sourceforge.net/projects/aicwl/files/releases/aicwl_3_25.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) :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadolib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
# Activation de la bibliothèque GTK-OSX avec pkg-config
% PATH=$xnadainst/bin:$PATH
# Activation de la bibliothèque GTKAda
% export GPR_PROJECT_PATH=$xnadainst/share/gpr
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% mkdir aicwl_3_25
% cd aicwl_3_25
% tar xzf $xnarchive/aicwl_3_25.tgz
# Récupérer les projets GPR apportés sur Blady
% unzip $xnarchive/xnadolib-2022-diff/aicwl_gpr.zip
% gprbuild -p -P lib_aicwl.gpr -XSC_OS=OSX
# Utilisation de gnatdoc de GNAT CE 2019
% /usr/local/adacore/2019/bin/gnatdoc --enable-build --no-subprojects -P lib_aicwl.gpr
-XSC_OS=OSX
% gprinstall -f -p --prefix=$xnadainst -P lib_aicwl.gpr -XSC_OS=OSX
% gprbuild -p -P xpm2gtkada/build_xpm2gtkada.gpr -XSC_OS=OSX
% gprinstall -f -p --prefix=$xnadainst -P xpm2gtkada/build_xpm2gtkada.gpr -XSC_OS=OSX
```

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

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

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

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

```
% cd $xnadainst/share/examples/aicwl
% gprbuild -p -P build_examples.gpr -XSC_OS=OSX
```

```
% export XDG_DATA_DIRS=$xnadainst/share
% ./bin/oscilloscope_plotter
...
```

7. Construire Templates-Parser

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

Site Web : github.com/AdaCore/templates-parser

La version installée est 22.2.

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 :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% git clone https://github.com/AdaCore/templates-parser.git -b 22.2
% cd templates-parser
% make prefix=$xnadainst setup
% make -w
# Utilisation de Sphinx et LaTeX avec MacPorts
% PATH=/opt/local/bin:$PATH make build-doc
% 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.

La version installée est 0.5d.

Dépendances : GTKAda, Templates Parser.

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) :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
```

```
% PATH=/opt/gcc-12.1.0/bin:$PATH
# Activation de la bibliothèque GTK-OSX avec pkg-config
% PATH=$xnadainst/bin:$PATH
# Activation des bibliothèques GTKAda et Templates Parser
% export GPR_PROJECT_PATH=$xnadainst/share/gpr
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% git clone https://github.com/Blady-Com/gate3.git -b V0.5d
% cd gate3
% make -w
% make PREFIX=$xnadainst 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

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

Les fichiers modèles pour la génération des sources Ada dans le dossier `$xnadainst/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

9. Construire Florist

Florist contient les composants conformement aux standards Posix Ada : IEEE Standards 1003.5: 1992, IEEE STD 1003.5b: 1996 et en partie IEEE STD 1003.5c: 1998.

Site Web : www.cs.fsu.edu/~baker/florist.html

La version installée est 2022a.

Dépendances : -.

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

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% git clone https://github.com/Blady-Com/florist.git -b xnadalib-2022
% cd florist
% ./configure --prefix=$xnadainst
% Build=Debug make ADAFLAGS=-gnatwn
# Utilisation de gnatdoc de GNAT CE 2019
% PATH=/usr/local/adacore/2019/bin:$PATH make rm-doc
% make install
```

Une documentation au format HTML est disponible dans le dossier `$xnadainst/share/doc/florist/florist_rm`.

10. Construire AdaCurses

AdaCurses est une bibliothèque Ada 95 basée sur NCurses. La correspondance avec les fonctions de NCurses n'est pas directe mais a été construite dans l'esprit Ada de privilégier la lisibilité.

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

La version installée est 6.3-20221105.

Dépendances : -.

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

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% git clone https://github.com/Blady-Com/ncurses.git -b xnadalib-2022
```

```
% cd ncurses
% ./configure CC='gcc -g -O0' --enable-pc-files --with-ada-sharedlib --with-shared --enable-symlinks
--prefix=$xnadainst --with-ada-libname=adacurses
% make -w
# Utilisation de gnatdoc de GNAT CE 2019
% PATH=/usr/local/adacore/2019/bin:$PATH make -C Ada95/src rm-docs
% make install
% make -C Ada95/src install-new
```

Des exemples de programme sont disponibles dans le dossier `$xnadainst/share/examples/adacurses`.

Une documentation au format HTML est disponible dans le dossier `$xnadainst/share/doc/adacurses`.

11. Construire Zanyblue

Zanyblue est une bibliothèque native en Ada pour l'internalisation d'un logiciel avec l'affichage de textes traduits dans différentes langues.

Site web : zanyblue.sourceforge.net.

Version installée : 1.4.0.

Dépendances : -.

Récupérer l'archive suivante sur le bureau :

sourceforge.net/projects/zanyblue/files/zanyblue-1.4.0.tar.gz

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) :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% tar xzf $xnarchive/zanyblue-1.4.0.tar.gz
% cd zanyblue-1.4.0
% cd src
% make -w
% make INSTALL_DIR=$xnadainst install
```

Une documentation au format PDF et HTML est disponible dans le dossier `$xnadainst/share/doc/zanyblue` :

```
% open $xnadainst/share/doc/zanyblue/ZanyBlue.pdf
% open $xnadainst/share/doc/zanyblue/index.html
% open $xnadainst/share/doc/zanyblue/ref/index.html
```

Des exemples de programme sont disponibles dans le dossier `$xnadainst/share/examples/zanyblue`.

12. Construire PragmARC

PragmAda Reusable Components (PragmARCs) est une collection de composants mathématiques, de dates, de listes proposée par Jeff Carter.

Site web : pragmada.x10hosting.com/pragmarc.htm.

La version installée est mid-2022.

Dépendances : -.

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) :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% git clone https://github.com/Blady-Com/PragmARC.git -b xnadalib-2022
% cd PragmARC
% gprbuild -p -P lib_pragmarc.gpr
# Utilisation de gnatdoc de GNAT CE 2019
% /usr/local/adacore/2019/bin/gnatdoc --enable-build -P lib_pragmarc.gpr
% gprinstall -f -p --prefix=$xnadainst lib_pragmarc.gpr
```

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

```
% open $xnadainst/share/doc/pragmarc/pragmarc_rm/index.html
```

13. Construire UXStrings

L'objectif de cette bibliothèque est de proposer en Ada un concept de chaînes de caractères de taille dynamique et avec des caractères Unicode. Les API sont inspirées de `Ada.Strings.Unbounded` afin de minimiser le travail de reprise de codes Ada existants.

Site Web : github.com/Blady-Com/UXStrings

La version installée est 0.4.0.

Dépendances : Simple Components.

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

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
# Activation de la bibliothèque Simple Components
% export GPR_PROJECT_PATH=$xnadainst/share/gpr
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% git clone https://github.com/Blady-Com/UXStrings.git -b xnadalib-2022
% cd UXStrings
% gprbuild -p -P lib_uxstrings.gpr
# Utilisation de gnatdoc de GNAT CE 2019
% /usr/local/adacore/2019/bin/gnatdoc --enable-build --no-subprojects -P lib_uxstrings.gpr
% gprinstall -f -p --prefix=$xnadainst lib_uxstrings.gpr
```

Une documentation au format HTML est disponible dans le dossier `$xnadainst/share/doc/uxstrings/uxstrings_rm` :

```
% open $xnadainst/share/doc/uxstrings/uxstrings_rm/index.html
```

14. Construire Gnoga

Gnoga est une bibliothèque graphique créée nativement en Ada. Ce n'est pas une sur-couche Ada à une bibliothèque existante en C ou C++. Sa particularité est de permettre de construire des applications graphiques orientées Web indépendantes de la plateforme. Double indépendance garantie d'une part de fait du langage Ada lui même, Ada assure qu'un code source aura un comportement identique quelque soit la plate-forme d'exécution de part son compilateur (s'il accepte la compilation), d'autre part avec l'utilisation du HTML, CSS et Javascript pour le rendu graphique dans un navigateur Web.

Source : www.gnoga.com.

La version installée est 2.2-mid-2022.

Dépendances : Simple Components, UXStrings.

Avant de démarrer la compilation, GNAT doit être activé, ajouter son emplacement comme par exemple :

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
# Activation de la bibliothèque Simple Components et UXStrings
% export GPR_PROJECT_PATH=$xnadainst/share/gpr
```

Saisir les commandes suivantes dans le Terminal :

```
% cd /usr/local/src-2022
% git clone https://github.com/Blady-Com/gnoga.git -b xnadalib-2022
% cd gnoga-code
% make html-docs
# Utilisation de gnatdoc de GNAT CE 2019
% make PREFIX=$xnadainst install_gnoga_sa GNATDOC=/usr/local/adacore/2019/bin/gnatdoc
% tar -cf - demo tutorial | tar -C $xnadainst/share/gnoga -xf -
```

Des démos et tutoriels sont présents respectivement dans les dossiers `$xnadainst/share/gnoga/demo` et `$xnadainst/share/gnoga/tutorial`.

Une documentation sous forme de manuels utilisateur et d'un manuel de référence est disponible dans le dossier `$xnadainst/share/gnoga/html`.

```
% open $xnadainst/share/gnoga/html/user_guide.html
% open $xnadainst/share/gnoga/html/api_summary.html
% open $xnadainst/share/gnoga/html/gnoga_rm/index.html
```

Voir l'utilisation de Gnoga avec des exemples sur Blady :
blady.pagesperso-orange.fr/a_savoir.html#gnoga

15. Construire SparForte

SparForte est un interpréteur du langage AdaScript, il peut servir de shell Ada ou de moteur de création de page HTML en AdaScript qui est un petit sous-ensemble du langage de programmation Ada.

Site web : sparforte.com.

Version installée : 2.5.

Dépendances : -.

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

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadalib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% git clone https://github.com/Blady-Com/SparForte.git -b xnadalib-2022
% cd SparForte
% ./configure --prefix=$xnadainst --manprefix=$xnadainst --without-bdb --without-mysql --without-opengl --without-postgres --without-readline --without-sound --arch=native --without-pcre --without-sdl
```

```
% make -w
% make install
```

Le manuel est dans le dossier `$xnadainst/man/man1` :

```
% MANPATH=$xnadainst/man man spar
```

L'interpréteur se lance ainsi:

```
% cd $xnadainst/bin
% ./spar
Type "help" for help
=> put_line ("Hello again with Ada!")
Hello again with Ada!
=> return
```

16. Construire HAC

HAC est un interpréteur du langage Ada, il peut servir de shell Ada ou de moteur de création de page HTML en AdaScript qui est un petit sous-ensemble du langage de programmation Ada.

Site web : hacadacompiler.sourceforge.io.

Version installée : 0.21 mid-2022.

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

```
% xnadabase=/usr/local
% version=2022
% xnadasrc=$xnadabase/src-$version
% xnadainst=$xnadabase/xnadolib-$version
% xnarchive=$HOME/Downloads
# Activation du compilateur Ada GNAT FSF 12.1
% PATH=/opt/gcc-12.1.0/bin:$PATH
```

Saisir les commandes suivantes dans le Terminal :

```
% cd $xnadasrc
% git clone https://github.com/zertovitch/hac.git
% cd hac
% gprbuild -P hac.gpr
% gprinstall -p --prefix=$xnadainst --mode=usage -P hac.gpr
```

L'interpréteur se lance ainsi :

```
% cd $xnadainst/bin
% ./hac
```

Pascal Pignard, avril-septembre 2011, août 2014, juillet 2015, août-septembre 2016, octobre 2017, septembre 2018, octobre 2021, novembre 2022.

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