

Compilation de l'IDE C-Workshop sous Windows

par Nicolas Joseph ([home](#))

Date de publication : 01 Juin 2005

Dernière mise à jour :

Comment compiler le projet **C-Workshop** sous Windows avec Dev-cpp

- I - Introduction
- II - LibXml2
- III - GtkSourceView
 - III-A - gnu-regex
 - III-B - GtkSourceView
- IV - C-Workshop
- V - Conclusion

I - Introduction

Avant de commencer à compiler, il faut vous assurez d'avoir tous les outils nécessaires. Pour cela, vous devez avoir :

- **Dev-cpp** : c'est la version 4.9.9.2 avec **Mingw** qui est utilisée pour cette article
- **GTK+** la version 2.6 est nécessaire ainsi que le devpack pour le kit de developpement (disponible sur [devpacks.com](#))
- **GtkSourceView** : testé uniquement avec la version 1.2.0
- **LibXml2** : le devpak contenant la LibXml2
- Et bien sûr les sources du projet disponible sur sourceforge et récupérable grâce à CVS : **C-Workshop**

Normalement l'installation de **Dev-cpp** ne doit pas poser de problème. Idem pour **GTK+** et son kit de developpement si vous utiliser le package manager de **Dev-cpp**. Par contre pour le reste, je vais vous exposer la marche à suivre.

Pour commencer décompressez le package contenant le projet, cela devrait créer un dossier *cworkshop* qui sera notre répertoire de travail (je considère par la suite qu'il s'agit de *c:\cworkshop*).

II - LibXml2

GtkSourceView nécessite l'installation de la **LibXml2** (1) il faut donc commencer par l'installer, tout simplement en utilisant le **Package Manager** de **Dev-cpp**. Ctrl + O pour sélectionner le paquetage à installer.

III - GtkSourceView

Commencez par extraire l'archive contenant **GtkSourceView** dans le répertoire de travail (*c:\cworkshop* dans notre cas). Avant de compiler les sources de **GtkSourceView** à proprement parlées, il faut compiler **gnu-regex** qui se trouve dans le même package.


III-A - gnu-regex

Il faut commencer par créer un nouveau projet nommé *libgnuregex* sous **Dev-cpp** en choisissant *Static library* comme type de projet et le C comme langage :

Enregistrer le projet dans le dossier des sources de **gnu-regex** :

Ensuite, il suffit d'ajouter le fichier *regex.c* au projet et d'ajouter le dossier *C:\cworkshop\gtksourceview-1.2.0\gtksourceview\gnu-regex* à la liste des répertoires d'inclusion pour cela, il suffit d'aller dans les Options du projet (Alt + P) :

Enfin il ne vous reste plus qu'à compiler la bibliothèque (Ctrl + F9).

 *Je vous conseille de ne pas ajouter de directive de compilation tel que **-ansi** ou encore les fameuses options de warning (**-Wall**, **-Werror**...) tout simplement parce que les bibliothèques compilées ici ne les respectent pas forcément : par exemple, la bibliothèque **gnu-regex** utilise le mot clés **inline**.*

III-B - GtkSourceView

Maintenant que nous avons *libgnuregex.a*, nous pouvons créer la bibliothèque statique *libgtksourceview.a* :

Projet à enregistrer dans le répertoire des sources de **GtkSourceView** :

Ensuite ajoutez tous les fichiers *.c* du répertoire *gtksourceview* au projet sauf le fichier *gtksourceprintjob.c* car il nécessite **gnome-print** pour compiler :


Dans les Options du projets (Alt + P), copier/coller les options de compilation et de linkage utilisées dans le cas d'un projet GTK+ (peut être trouvées dans le fichier template de **Dev-cpp** du package **GTK+**) :

```
-I"<INCLUDE>\gtk-2.0"  
-I"<LIB>\gtk-2.0\include"  
-I"<INCLUDE>\atk-1.0"  
-I"<INCLUDE>\pango-1.0"  
-I"<INCLUDE>\glib-2.0"  
-I"<INCLUDE>\glib-2.0\glib"  
-I"<LIB>\glib-2.0"  
-I"<LIB>\glib-2.0\include"  
-mno-cygwin
```

```
-mms-bitfields
```

Pendant que nous sommes dans les options du compilateur, il va falloir bidouiller un peu, en effet les sources contiennent des macro qui sont définies lors d'une procédure standard d'installation sous Linux (./configure && make && make install) mais comme nous n'utilisons pas ce système de génération, il faut ajouter ceci aux directives de compilation du compilateur :

```
-DLOCALEDIR="\ /locale\ " "  
-DGETTEXT_PACKAGE="\ gtksourceview-1.0\ " "  
-DDATADIR="\ /usr/share/\ " "  
-DENABLE-NLS
```

 *J'ai essayé de retrouver ces valeurs en parcourant les fichiers Makefile.am donc elles sont sûrement incorrectes mes permettent de compiler le projet. Il semble qu'elles ne soit utilisée qu'en cas d'erreur par conséquent, l'utilisation normale du programme ne devrait pas être modifiée.*

Encore une fois, il faut ajouter un dossier à la liste des répertoires d'inclusion, il s'agit du dossier racine de l'archive **GtkSourceView** :

Ctrl + F9, on obtient un fichier *libgtksourceview.a*, maintenant passons à la compilation du projet lui même!

IV - C-Workshop


Dernière ligne droite avant de pouvoir utiliser notre IDE! Créons un dernier projet, mais cette fois il s'agit d'un projet **GTK+** :

Projet que nous enregistrons dans le répertoire racine du projet :

Ensuite, il faut ajouter les fichiers sources au projet, parcourez les différents dossiers des modules et ajouter le contenu de chaque sous-dossier *src*, avec la version utilisée pour cet article, voici le résultat :

Chaque module ayant ses propres fichiers d'en-tête, il faut ajouter tous les sous-dossiers *inc* à la liste des répertoires d'inclusion et aussi le dossier contenant les en-têtes de **GtkSourceView** :

A présent, il ne reste plus qu'à résoudre les problèmes liés à l'édition des liens. Dans la fenêtre des options de l'éditeur de liens, cliquez sur le bouton *Ajouter fichier* et ajouter les deux bibliothèques précédemment créées ainsi que le fichier *libxml2.dll* présent dans le répertoire *bin* de **Dev-cpp** :

 *Il faut copier/coller les noms des fichiers ajoutés avant les options de la ligne de commande sinon vous obtenez des erreurs de linkage.*

Pour que le programme trouve les ressources dont il a besoin, il doit être placé dans le répertoire *bin*, toujours dans la fenêtre des options du projet, mais cette fois dans l'onglet *Construction*, il faut spécifier *bin* comme *Répertoire de sorties des Executables* :

Appuyer une dernière fois sur les touches **Ctrl + F9** pour compiler le projet.

V - Conclusion

Pour pouvoir exécuter **C-workshop**, vous devez copier/coller *libxml2.dll* dans le même répertoire que le programme. Voici le résultat :

Téléchargez l'archive contenant tous les fichiers nécessaire à la compilation du projet C-Workshop : **archive ZIP**

1 : Pour ceux que le langage XML intéresse, je vous invite à visiter la **rubrique XML de Developpez**