

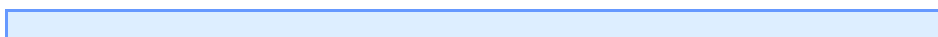
Installer SDL sous C++ builder



par [Loka](#)

Date de publication : 01/05/2006

Dernière mise à jour : 01/05/2006





L'installation de SDL pour Borland C++ 5 et 6 sur Windows est un peu plus difficile que sur un autre IDE comme Code::Blocks ou DevC++.

Cependant ce n'est pas irréalisable, je vais vous expliquer comment procéder.

Pour commencer, vous devez posséder un petit utilitaire du nom de **IMPLIB** qui est un outil en ligne de commande fournit par Borland.

Ce petit utilitaire vous permet de générer une lib à partir d'une dll (et donc de générer une lib compatible avec Borland...).

Son utilisation est très simple, il vous suffit de taper ceci :

```
implib.exe -a -c nom nom.dll
```

Ainsi vous aurez généré la nom.lib à partir de nom.dll.

Occupons nous de SDL maintenant.

Premièrement, il va falloir tout d'abord générer SDL.lib à partir de SDL.dll, il vous suffit de faire comme indiqué au dessus.

Ensuite, vous avez toujours besoin du fichier **sdl_main.lib**, de sorte que vous n'avez pas besoin de définir WinMain() et tout les autres trucs Windows.

Vous n'avez aucun *.dll à construire depuis cette lib, et COFF2OMF vous permet de créer une lib vide (que vous pouvez vérifier avec TDUMP, un autre utilitaire du compilateur).

Obtenez la version Win32 du fichier source SDL_main.c, et mettez-la avec le reste de votre projet.

Compilez votre projet (incluant **SDL_main.c**) en un fichier objet en utilisant BCC32 avec les options de compilation suivantes :

```
-c -tW -dWIN32
```

-c permet de dire de ne pas faire les liens.

-tW crée une application Windows.

-dWIN32 définit WIN32, qui est nécessaire dans le fichier SDL_main.h, et sans doute d'autres.

Il reste à faire la liaison entre les fichiers objet créés en utilisant ILINK32 avec les options suivantes :

-aa (créé une application Windows)

-Tpe (target = Windows exe)

-c (sensible à la casse à l'édition de lien)

Et les fichiers additionnels suivant (attention, l'ordre est important) :

* sdl.lib (créé au début)

* import32.lib

* c0w32.obj

* cw32.lib

Si tout s'est bien passé, vous devriez avoir un exécutable qui marche.