

# Windows平台搭建Code Blocks 17.12 + wxWidgets 3.1.x 开发环境

## 1. 安装Code Blocks

下载CodeBlocks IDE，选择自带MinGW的版本，目前最新版是：codeblocks-17.12mingw-setup.exe，下载后安装。

安装完成后，注意查看MinGW是否已加入系统环境变量。方法为在cmd下执行它包含的工具，例如，可在cmd下执行：

```
mingw32-make -v
```

如果能正常输出类似以下信息

```
GNU Make 3.82.90
Built for i686-pc-mingw32
Copyright (C) 1988-2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

则说明Code Blocks及其包含的MinGW已成功安装。若无此信息，可尝试手动将MinGW目录添加到系统环境变量。MinGW的位置在：(Code Blocks安装目录)\MinGW\bin

## 2. 编译wxWidgets

下载wxWidgets，链接：<https://www.wxwidgets.org/downloads/>，注意下载 3.1.x 版本，只有此版本与 Code Blocks 17.12 兼容。建议下载其源码，自行编译。目前最新版为 3.1.2

解压下载的压缩包，如，解压至 C:\wxWidgets-3.1.2

开始编译之前，请在系统环境变量里移除Cygwin及MSYS等软件的路径，否则会产生冲突导致编译出错。可选择编译完成后恢复这些配置，这些配置只对编译wxWidgets有影响，对后续基于wxWidgets的开发无影响。

在cmd中定位到wxWidgets编译目录，如：

```
C:\wxwidgets-3.1.2\build\msw
```

至此已做好编译前的准备工作

清理工程

```
mingw32-make -f makefile.gcc clean
```

编译Release版本的完整动态库

```
mingw32-make -f makefile.gcc MONOLITHIC=1 SHARED=1 UNICODE=1 BUILD=release
```

参数说明:

**MONOLITHIC:** (adj. 整体的; 巨石的, 庞大的; 完全统一的)

控制是构建一个单一的库 (MONOLITHIC=1) 还是多个组件库 (MONOLITHIC=0)。使用单一构建, 会编译一个完整的wxWidgets库, 项目的设置和开发会更加简单, 缺点就是生成的可执行文件会更大。建议开发阶段使用单一构建, 待发布时, 再使用多组件库并去掉不必要的组件来减小可执行文件的大小。

- wxWidgets的单一构建仅会创建一个wxWidgets导入库 (如libwxmsw31.a) 以及一个DLL (如wxmsw312\_gcc\_custom.dll)。注: msw表示MS Windows, 312表示wxWidgets的版本, 由此可见, 不同平台不同版本编译结果命名会有对应的差异。
- wxWidgets的多库 (multilib) 构建会创建多个导入库 (libwx31\_base.a等) 以及多个DLL文件。
- 无论何种wxWidgets构建, 都会创建额外的静态库 (如libwxexpat.a、libwxjpeg.a等)。这些库对于wxWidgets的DLL构建一般是不需要的, 但是当使用静态构建的时候, 则是必须的。

**SHARED:**

控制wxWidgets是构建DLL (SHARED=1) 还是静态库 (SHARED=0)。利用构建的DLL, 主程序构建时间较快, 可执行文件更小。但发布时必须同时携带DLL文件。使用静态库可以做到只发布一个单独的.exe便可正常运行, 缺点是开发时编译及链接速度较慢。

- wxWidgets的DLL构建会创建导入库 (如 libwxmsw31.a) 以及DLL文件 (如wxmsw312\_gcc\_custom.dll)。你必须在发布你的程序的时候包含这个DLL。
- wxWidgets的静态构建只会创建静态库 (如 libwxmsw31.a), 发布的时候也无须包含wxWidgets的DLL。

**UNICODE:**

控制wxWidgets以及你的程序是否使用支持Unicode的宽字符串。建议打开这个选项, 方便支持其它语言。

- wxWidgets的Unicode (UNICODE=1) 构建将会创建带有“u”后缀的库, 例如: "libwxmsw31u.a", "wxmsw312u\_gcc\_custom.dll"。
- wxWidgets的Unicode构建会在wxWidgets库的输出目录中创建"mswu"或"mswud"目录(d表示debug版), 使用非Unicode(ANSI)会创建"msw"或"mswd"。
- wxWidgets的ANSI (UNICODE=0) 构建创建的库没有“u”后缀, 例如: "libwxmsw31.a", "wxmsw312\_gcc\_custom.dll"。

**BUILD:**

控制wxWidgets生成的版本: 调试版本 (BUILD=debug) 或者是发布版本 (BUILD=release)。绝大多数情况下你只需要wxWidgets的发布版本就可以了, 因为你不应该不想要去调试wxWidgets自身, 同时你依然可以通过链接wxWidgets的发布版本来构建你自己的程序的调试版本。

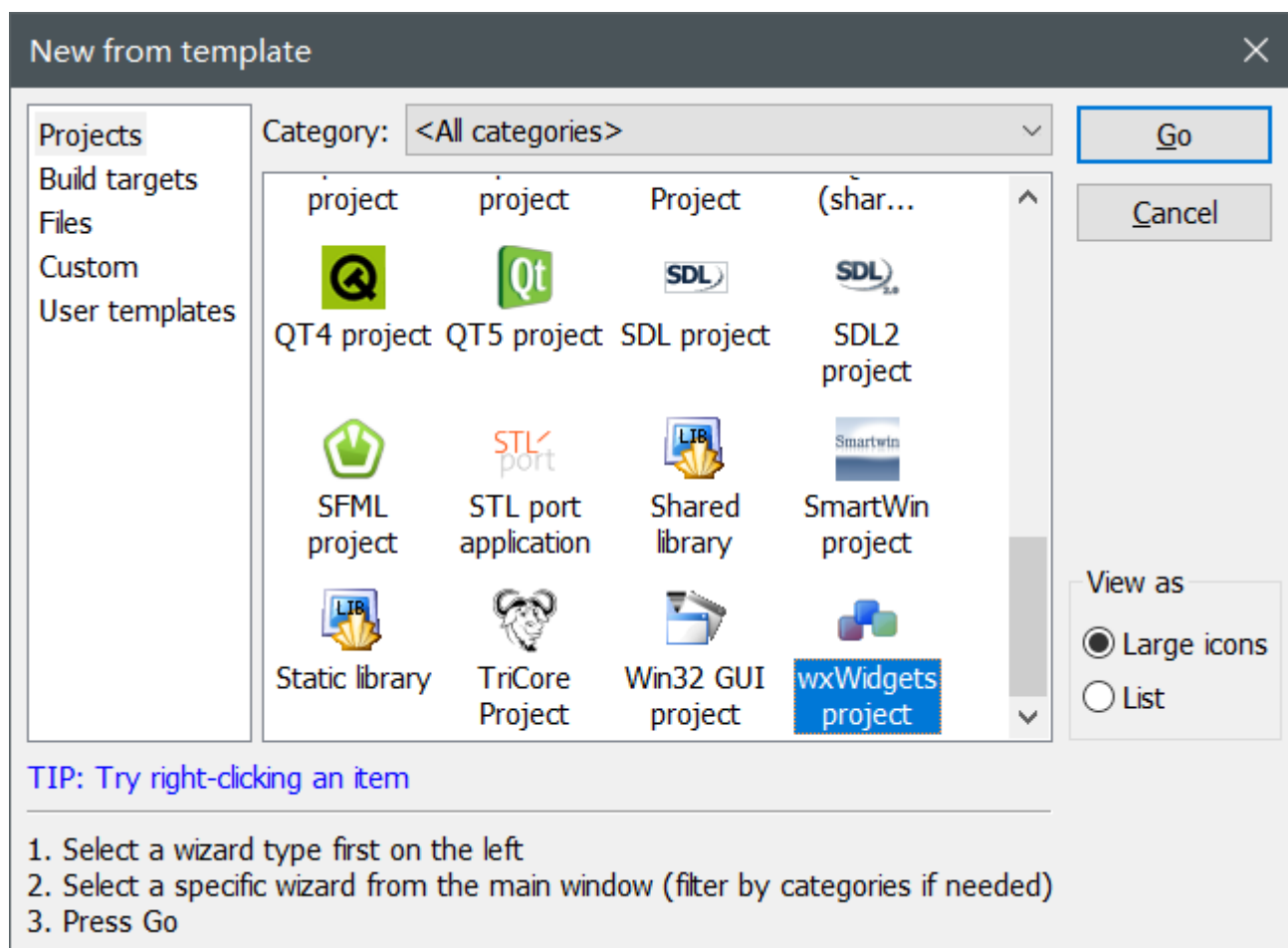
- 调试构建wxWidgets会创建带有“d”后缀的库, 例如“libwxmsw31d.a”、“wxmsw312d\_gcc\_custom.dll”。
- 调试构建wxWidgets会在wxWidgets库的输出目录中创建“mswd” 或者 “mswud” 目录。
- 发布构建wxWidgets创建的库没有“d”后缀, 例如“libwxmsw31.a”、“wxmsw312\_gcc\_custom.dll”。
- 发布构建wxWidgets会在wxWidgets库的输出目录中创建“msw” 或者 “mswu” 目录。

编译会耗费较长时间，请耐心等待。完成后，会在wxWidgets-3.1.2\lib目录下生成相关编译结果。编译的动态库输出至wxWidget目录的lib/gcc\_dll下，静态库则为lib/gcc\_lib。

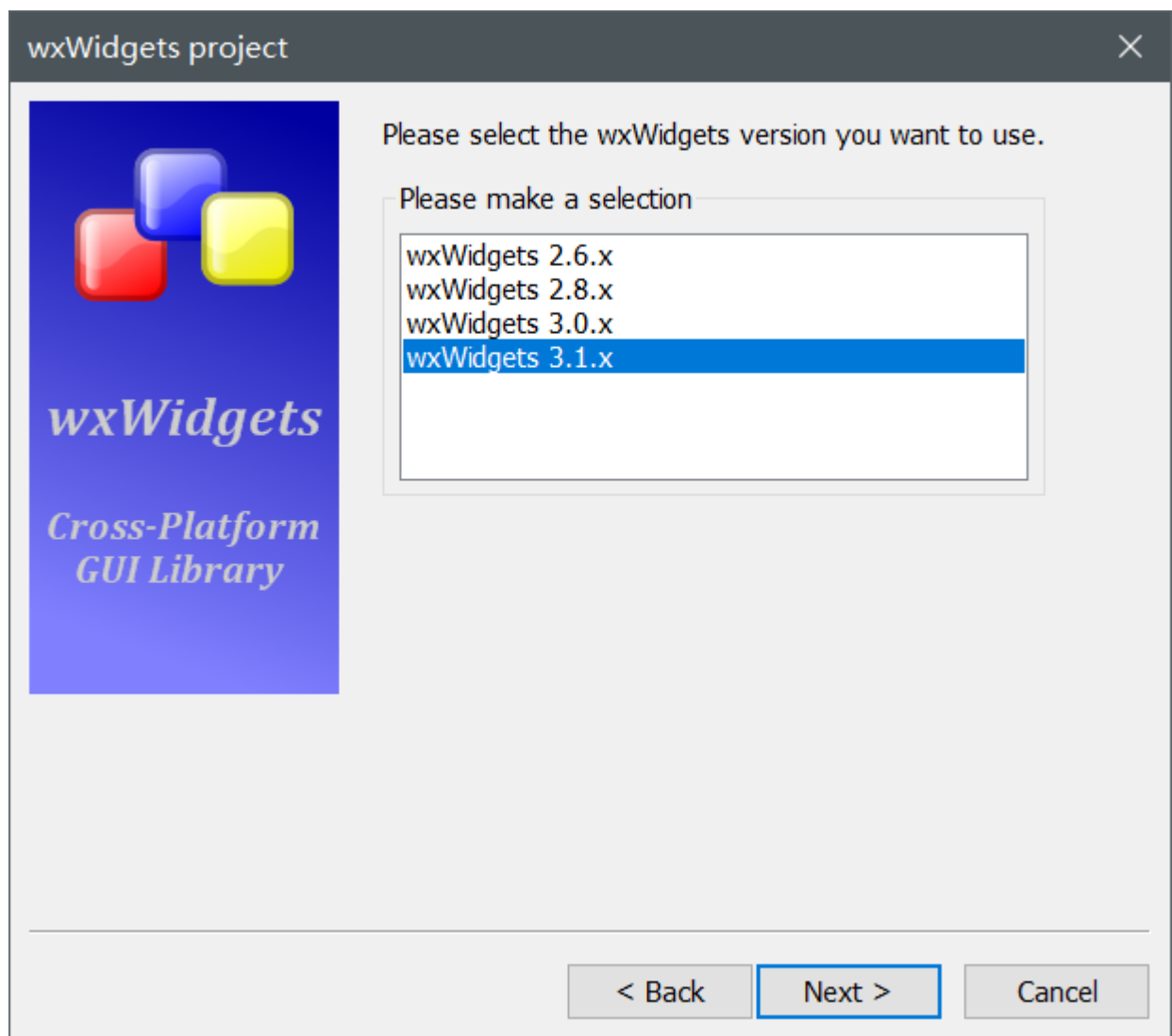
若有其它问题，请查阅：[https://wiki.wxwidgets.org/Compiling\\_wxWidgets\\_with\\_MinGW](https://wiki.wxwidgets.org/Compiling_wxWidgets_with_MinGW)

### 3. 工程搭建

打开Code Blocks，选择 File -> New -> Project，新建一个wxWidgets工程



点击Go开始工程向导，选择wxWidgets版本，这里我们选3.1.x



输入工程名

wxWidgets project

Please select the folder where you want the new project to be created as well as its title.

Project title:  
wxWidgetsDemo

Folder to create project in:  
D:\Test\

Project filename:  
wxWidgetsDemo.cbp


Resulting filename:  
D:\Test\wxWidgetsDemo\wxWidgetsDemo.cbp

*wxWidgets*  
*Cross-Platform*  
*GUI Library*

< Back   Next >   Cancel

输入作者相关信息，此部分内容会出现在Code Block自动生成文件的版权声明处(文件开头的注释)，建议认真添写

wxWidgets project



*wxWidgets*  
*Cross-Platform  
GUI Library*

Please Enter Project Details. Details will be used in labelling source code of project.

Author:

brant

Author's email:

bairixin2006@163.com

Author's website:

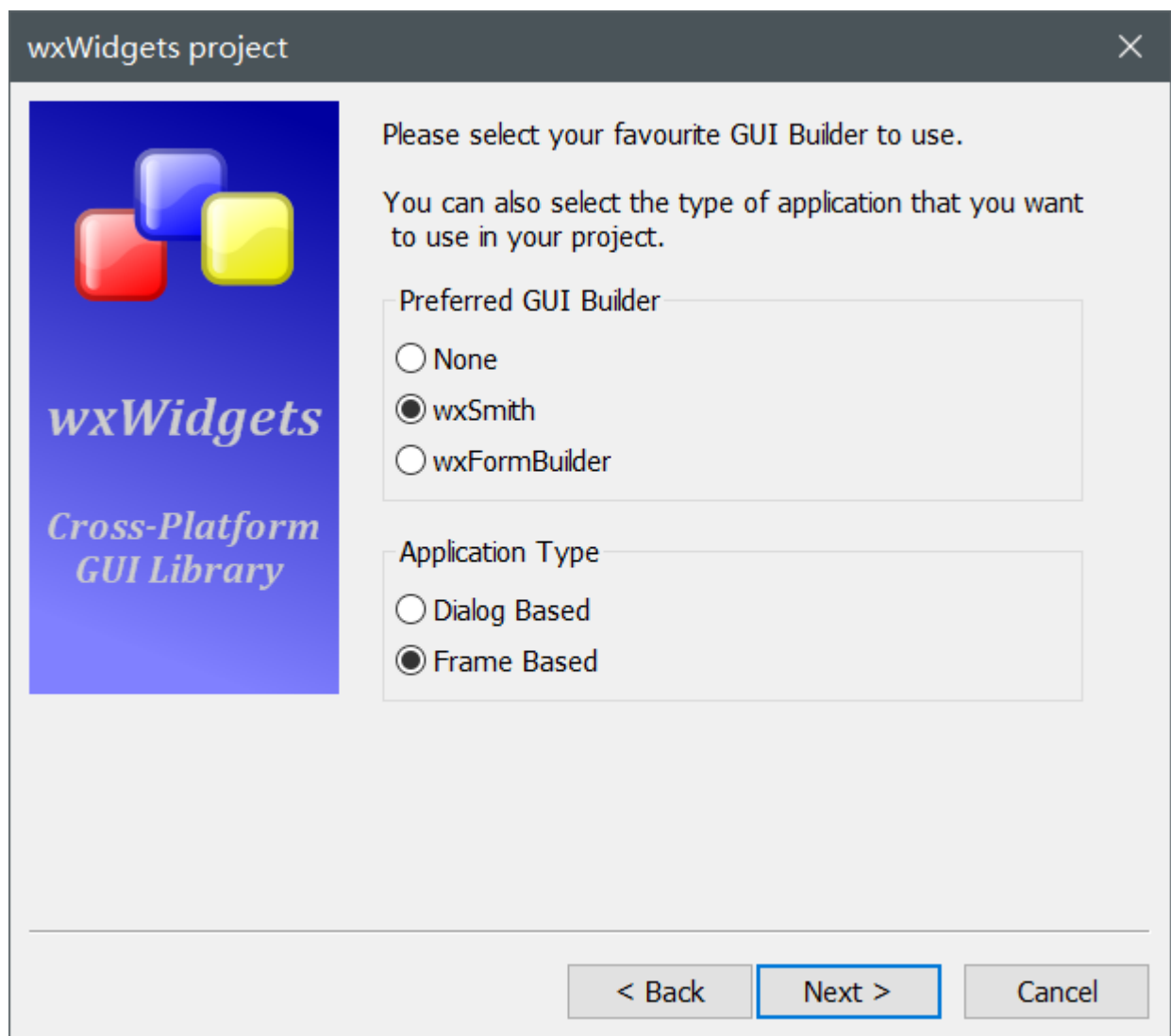
None

< Back

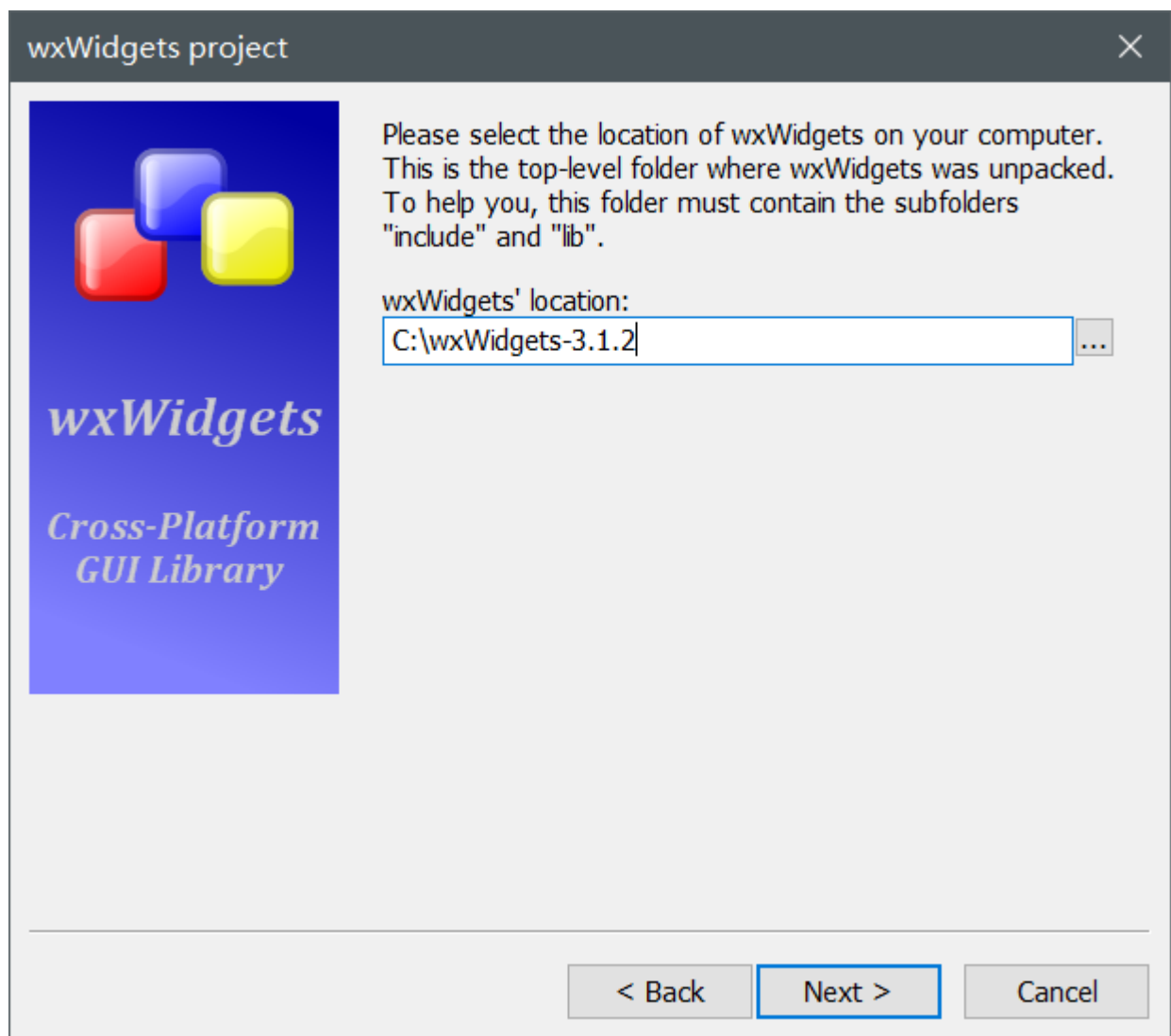
Next >

Cancel

选择UI编辑工具和UI类型，基于对话框还是基于框架



选择wxWidgets的目录，主要是为了查找include与lib，这里我们选根目录就好



选择编译器及Debug和Release的配置，使用默认就好，略过不提



wxWidgets project

Please select the compiler to use and which configurations you want enabled in your project.

Compiler:  
GNU GCC Compiler

☒ Create "Debug" configuration: Debug

"Debug" options

Output dir.: bin\Debug\  
Objects output dir.: obj\Debug\

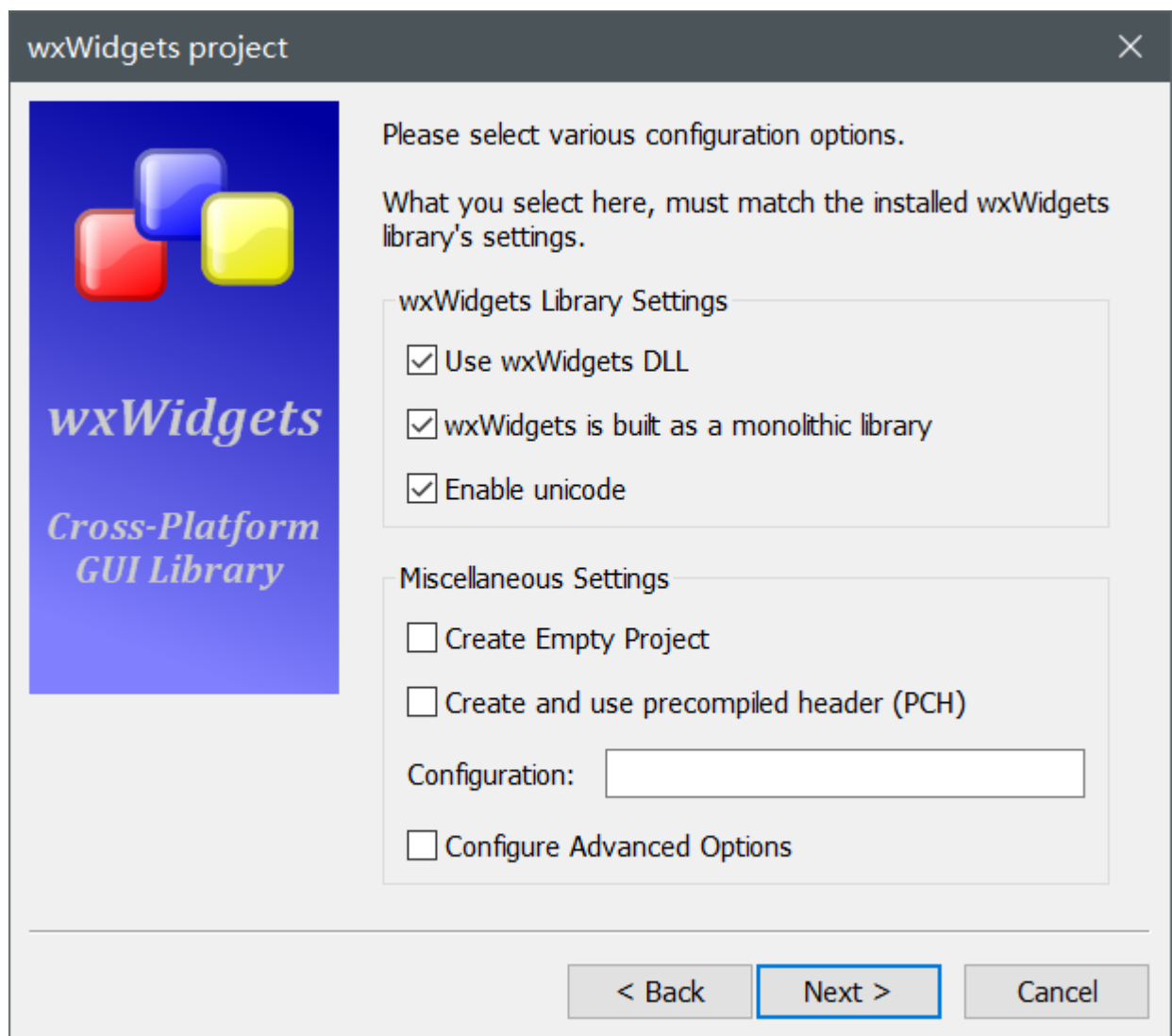
☒ Create "Release" configuration: Release

"Release" options

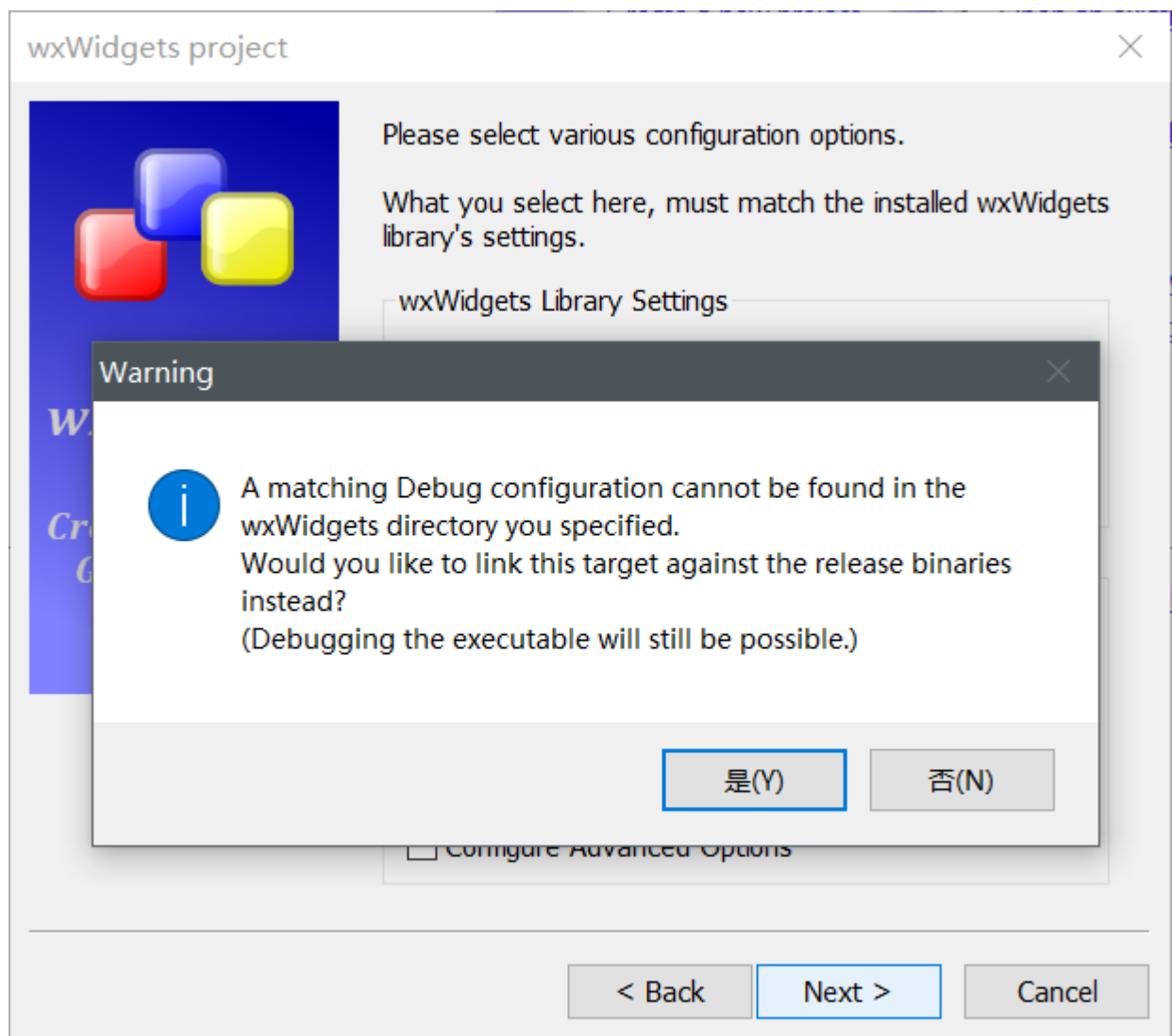
Output dir.: bin\Release\  
Objects output dir.: obj\Release\

< Back   Next >   Cancel

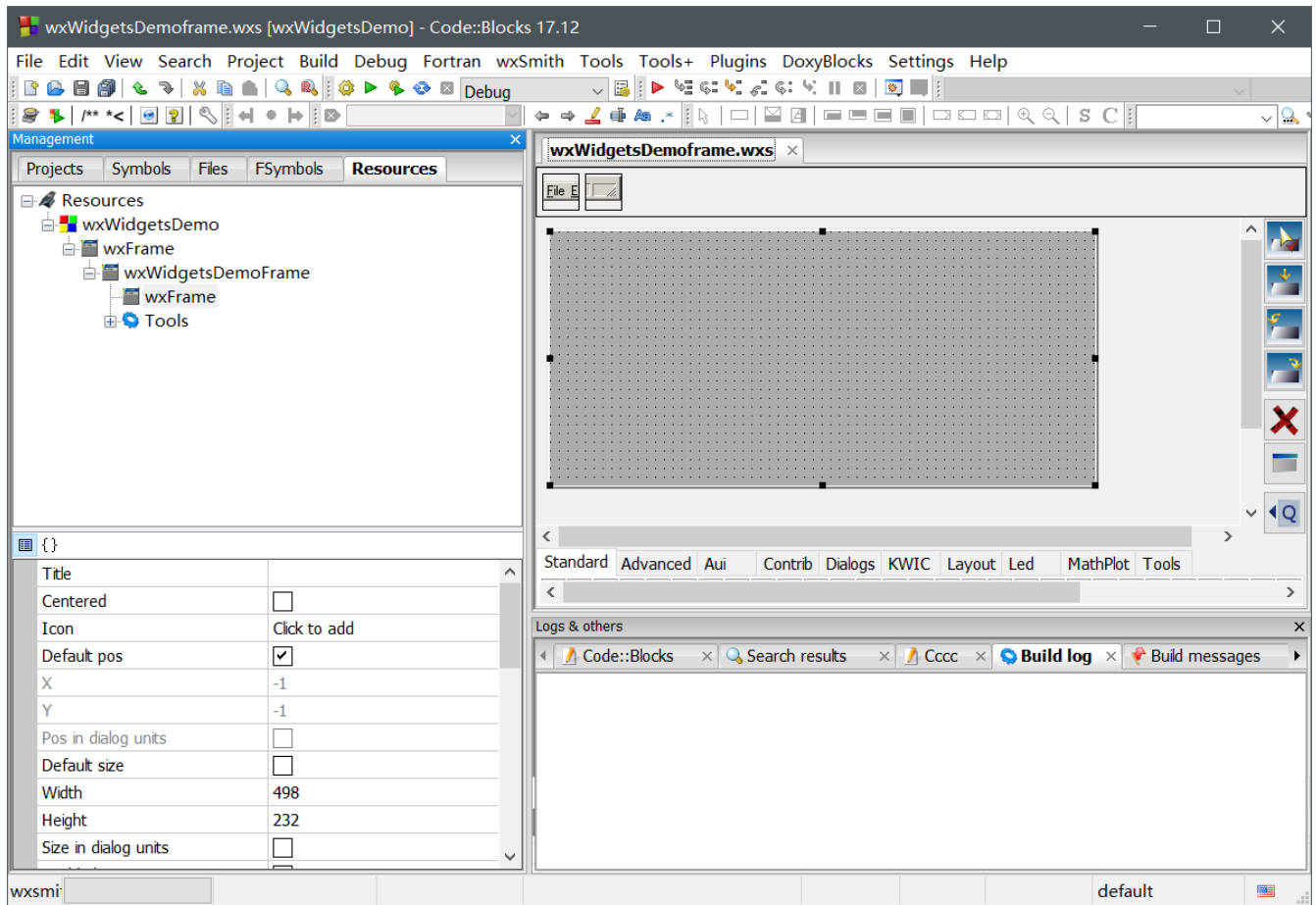
选择wxWidgets的配置，需要与上一步编译时输入的参数保持一致



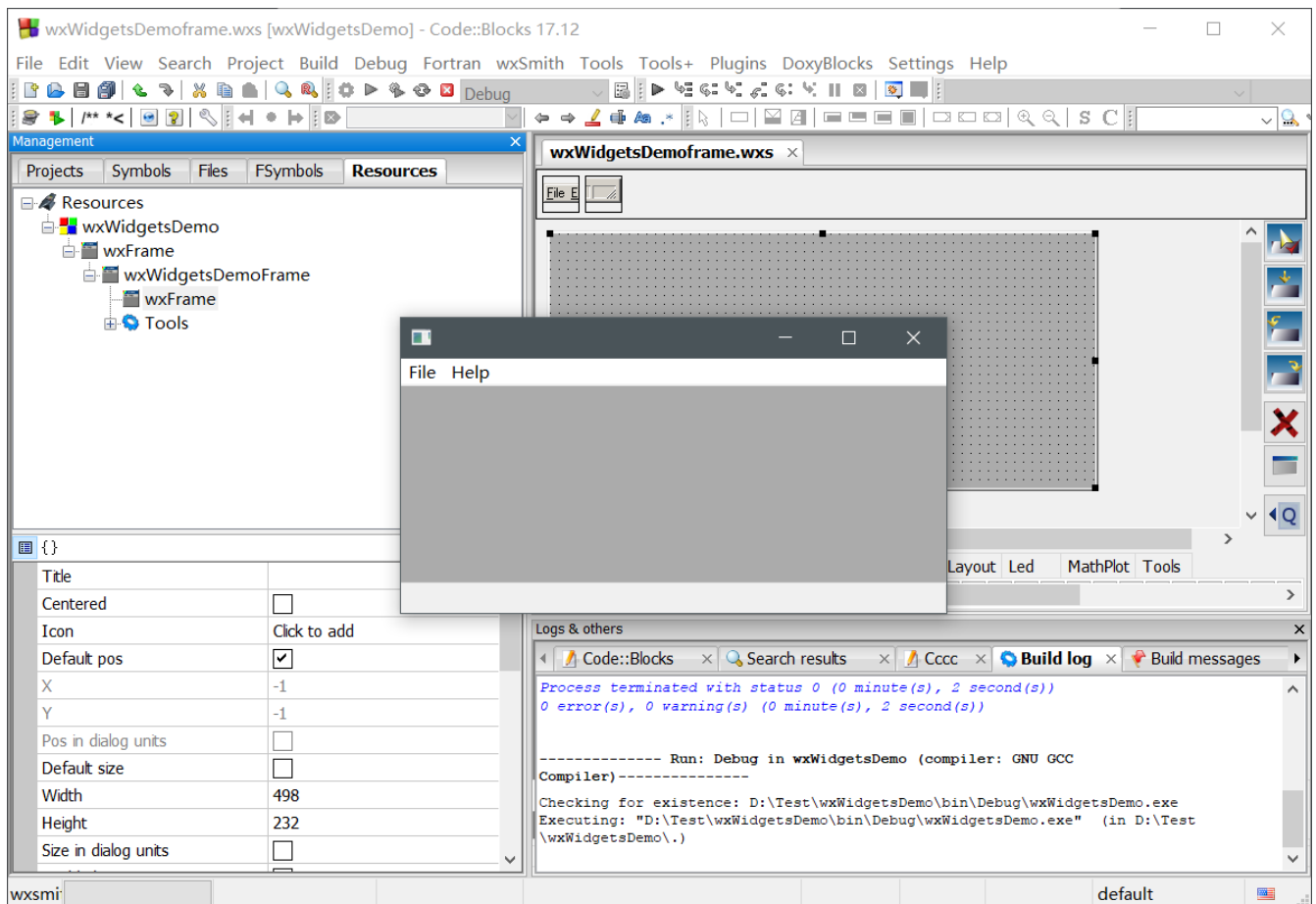
下一步后，若没有编译Debug版的wxWidgets，会有一个警告，大意是未找到Debug版的wxWidgets lib，若我们不想调试wxWidgets本身，忽略它即可。使用Release版的lib同样可以调试我们的程序



完成后，可以看到向导为我们创建的工程



## 编译运行



成功!