

# Настройка RedPanda-CPP для сборки проектов на wxWidgets

## 1. Сборка wxWidgets

Перед использованием wxWidgets в проекте необходимо собрать его из кода вручную. Для этого используется скрипт `wxWidgets\_win\_build(RPCpp).bat`.

### Процесс сборки скриптом:

1. Устанавливает путь к wxWidgets и MinGW
2. Добавляет путь к компилятору
3. Запускает `mingw32-make.exe` для сборки wxWidgets
4. Компилирует как `release`, так и `debug` версии wxWidgets

### Содержимое скрипта:

```
@echo off

:: Устанавливаем путь к wxWidgets
set WX_DIR=D:\Development\RedPanda-CPP\wxWidgets
:: Устанавливаем путь к платформе C++
set MINGW_DIR=D:\Development\RedPanda-CPP\mingw64
:: Устанавливаем имя утилиты сборки
set MAKE_TOOL=mingw32-make.exe

:: Устанавливаем путь сборки (НЕ МЕНЯЙТЕ ЭТО!)
set BUILD_DIR=%WX_DIR%\build\msw
:: Добавляем путь к компилятору в переменную PATH
set PATH=%MINGW_DIR%\bin;%PATH%

:: Переходим в папку для сборки
cd /d %BUILD_DIR%

:: Сборка релизной версии
echo Building release version...
%MAKE_TOOL% -f makefile.gcc BUILD=release SHARED=0 UNICODE=1
::%MAKE_TOOL% -j %NUMBER_OF_PROCESSORS% -f makefile.gcc BUILD=release
SHARED=0 UNICODE=1

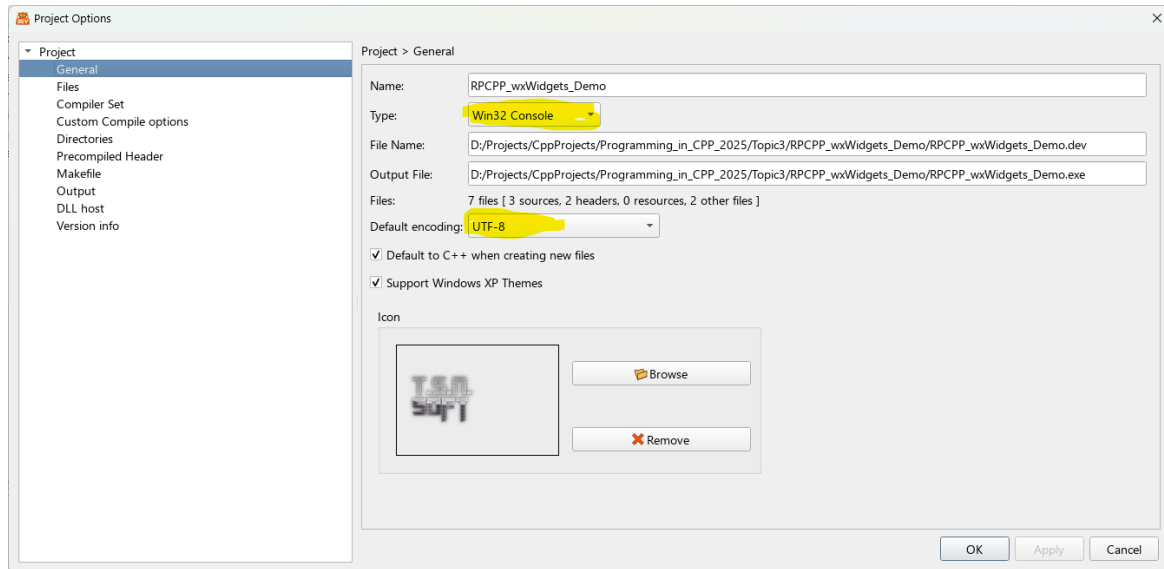
:: Сборка отладочной версии
echo Building debug version...
%MAKE_TOOL% -f makefile.gcc BUILD=debug SHARED=0 UNICODE=1
::%MAKE_TOOL% -j %NUMBER_OF_PROCESSORS% -f makefile.gcc BUILD=release
SHARED=0 UNICODE=1

:: Возвращаемся в исходную папку
cd /d %~dp0
```

## 2. Настройки проекта в RedPanda-CPP

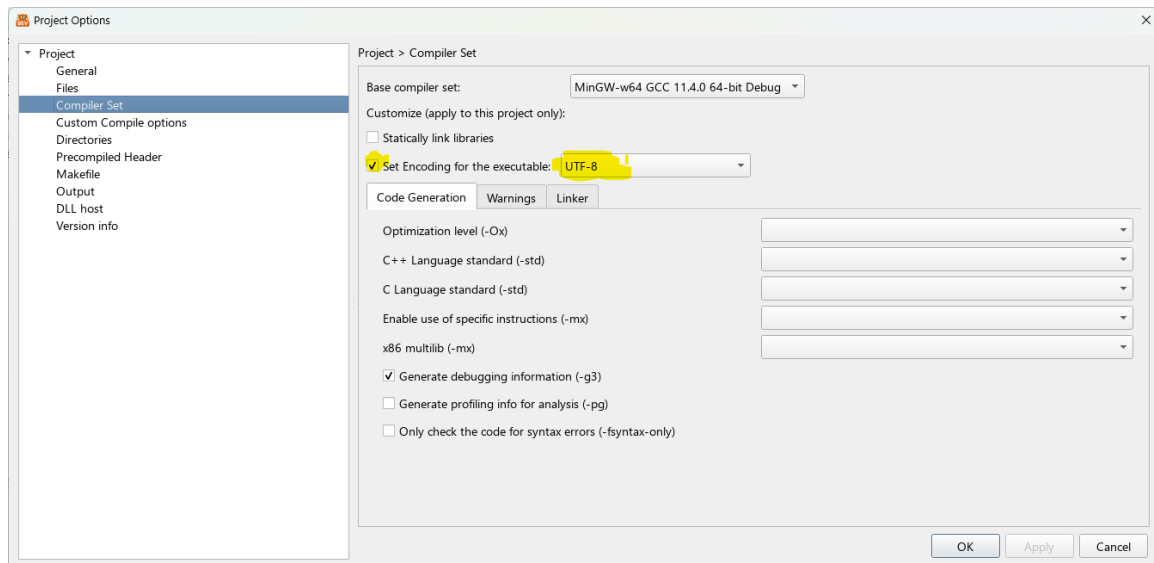
### 2.1 Общие параметры

- Тип проекта: `Win32 Console`
- Кодировка по умолчанию: `UTF-8`



### 2.2 Компилятор

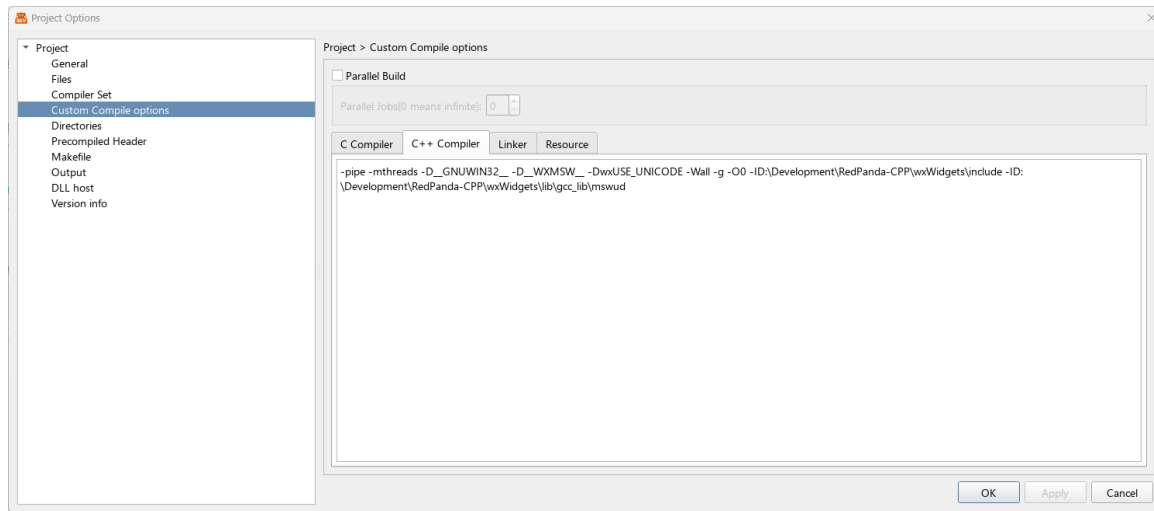
- Компилятор: `MinGW-w64 GCC 11.4.0 64-bit Debug`
- [✓] Set Encoding for the executable: `UTF-8`



## 2.3 Пользовательские опции компиляции

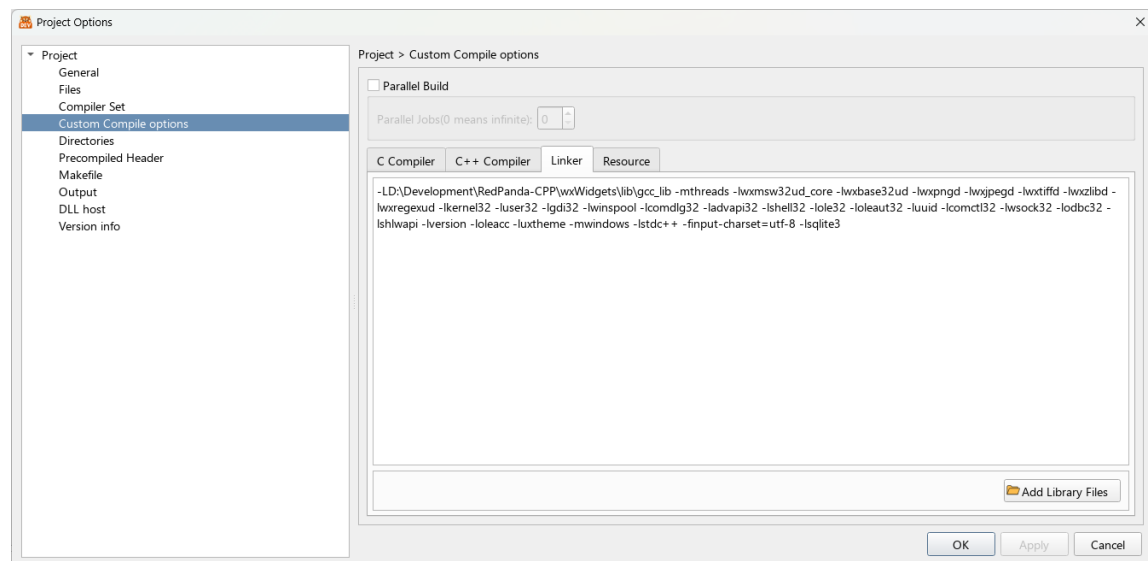
### C++ Compiler:

```
-pipe -mthreads -D_GNUWIN32_ -D_WXMSW_ -DwxUSE_UNICODE -Wall -g -O0  
-ID:\Development\RedPanda-CPP\wxWidgets\include  
-ID:\Development\RedPanda-CPP\wxWidgets\lib\gcc_lib\mswud
```



### Linker:

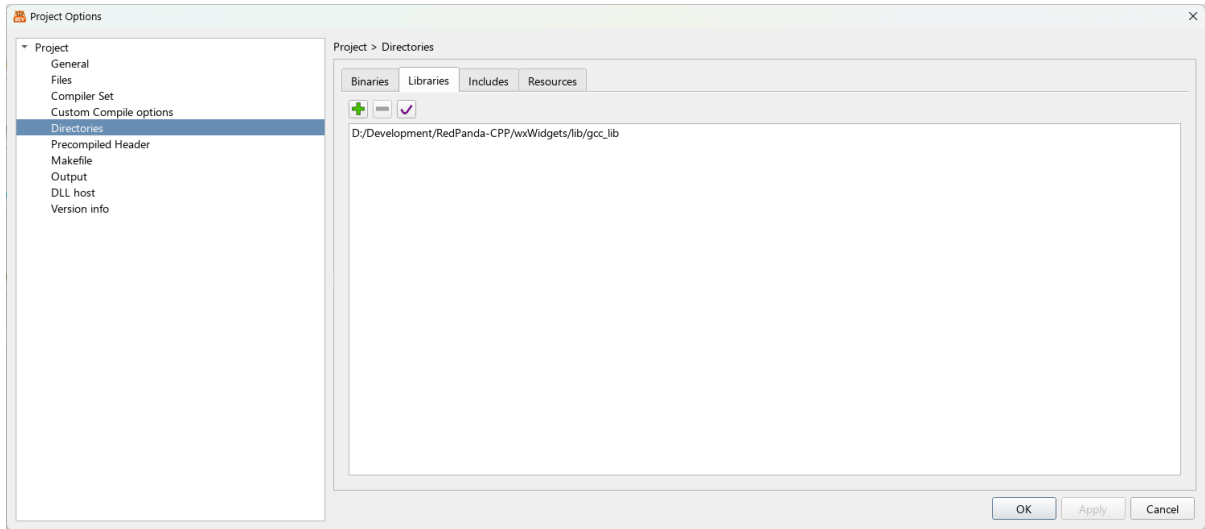
```
-LD:\Development\RedPanda-CPP\wxWidgets\lib\gcc_lib -mthreads  
-lwxmsw32ud_core -lwxbase32ud -lwxpngd -lwxjpegd -lwxzlibd  
-lwxregexud -lkernel32 -luser32 -lgdi32 -lwinpool -lcomdlg32 -ladvapi32  
-lshell32 -lole32 -loleaut32 -luuid -lcomctl32 -lwsck32 -lodbc32 -lshlwapi  
-lversion -loleacc -luxtheme -mwindows -lstdc++ -finput-charset=utf-8 -lsqllite3
```



## 2.4 Директории

### *Libraries:*

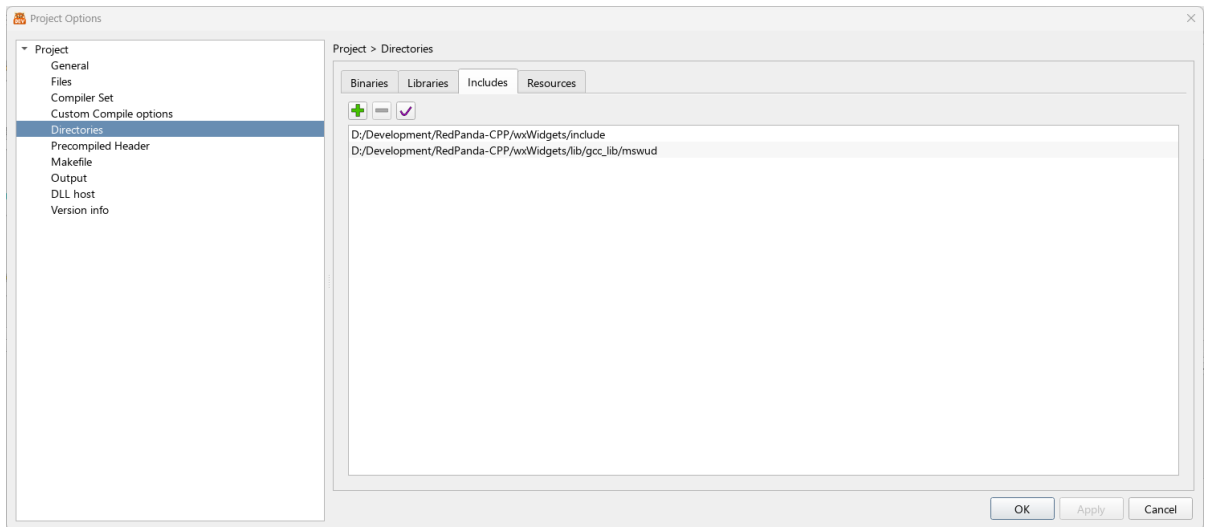
D:/Development/RedPanda-CPP/wxWidgets/lib/gcc\_lib



### *Includes:*

D:/Development/RedPanda-CPP/wxWidgets/include

D:/Development/RedPanda-CPP/wxWidgets/lib/gcc\_lib/mswud



## Дополнительные скрипты для сборки готовых программ

### 1. Скрипт сборки автономной Debug-версии готовой программы

```
@echo off

:: Устанавливаем переменную для основной папки (можно изменить на любой
путь)
set DEV_DIR=D:\Development\RedPanda-CPP

:: Добавляем путь к папке bin в переменную окружения PATH
set PATH=%DEV_DIR%\mingw64\bin;%PATH%

:: Устанавливаем пути для упрощения
set WX_DIR=%DEV_DIR%\wxWidgets

:: Указываем нужные библиотеки
set LIBS=-lwxmsw32ud_core -lwxbase32ud -lwxpngd -lwxjpegd -lwxtifff -
lwxzlibd -lwxregexud ^
-lkernel32 -luser32 -lgdi32 -lwinpool -lcomdlg32 -ladvapi32 -
lshell32 -lole32 -loleaut32 ^
-luuid -lcomctl32 -lwsck32 -lodbc32 -lshlwapi -lversion -
loleacc -luxtheme

:: Удаляем старый .exe файл если есть
if exist myappd.exe del myappd.exe

:: Компиляция с флагами
echo Compiling and assembling the program (static,debug) ...
g++ -o myappd.exe *.cpp ^
-I%WX_DIR%\include ^
-I%WX_DIR%\lib\gcc_lib\mswud ^
-L%WX_DIR%\lib\gcc_lib ^
-mthreads ^
%LIBS% ^
-g ^
-O0 ^
-static ^
-finput-charset=utf-8 ^
-mwindows ^
-lstdc++ ^
-lsqlite3

:: Проверяем, успешно ли был собран .exe файл
if exist myappd.exe (
    echo Compilation completed successfully. Launching program ...
    start /b myappd.exe
) else (
    echo Compilation error !
)
```

## 2. Скрипт сборки автономной Release-версии готовой программы

```
@echo off

:: Устанавливаем переменную для основной папки (можно изменить на любой
путь)
set DEV_DIR=D:\Development\RedPanda-CPP

:: Добавляем путь к папке bin в переменную окружения PATH
set PATH=%DEV_DIR%\mingw64\bin;%PATH%

:: Устанавливаем пути для упрощения
set WX_DIR=%DEV_DIR%\wxWidgets

:: Указываем нужные библиотеки
set LIBS=-lwxmsw32u_core -lwxbase32u -lwxpng -lwxjpeg -lwxrt -lwxzlib -lwxregexu ^
-lkernel32 -luser32 -lgdi32 -lwinsock -lcomctl32 -ladvapi32 -lshell32 -lole32 -loleaut32 ^
-luuid -lcomctl32 -lws2_32 -lshlwapi -lversion -lole32 -luxtheme

:: Удаляем старый .exe файл если есть
if exist myapp.exe del myapp.exe

:: Компиляция с флагами
echo Compiling and assembling the program (static,release) ...
g++ -o myapp.exe *.cpp ^
-I%WX_DIR%\include ^
-I%WX_DIR%\lib\gcc_lib\mswu ^
-L%WX_DIR%\lib\gcc_lib ^
-mthreads ^
%LIBS% ^
-static ^
-O2 ^
-DNDEBUG ^
-finput-charset=utf-8 ^
-mwindows ^
-lstdc++ ^
-lsqlite3

:: Проверяем, успешно ли был собран .exe файл
if exist myapp.exe (
    echo Compilation completed successfully. Launching program ...
    start /b myapp.exe
) else (
    echo Compilation error !
)
```

## Тестовый пример оконной wx-программы

```
#include <wx/wx.h>

class MyApp : public wxApp {
public:
    virtual bool OnInit();
};

class MyFrame : public wxFrame {
public:
    MyFrame();
};

wxIMPLEMENT_APP(MyApp);

bool MyApp::OnInit() {
    MyFrame *frame = new MyFrame();
    frame->Show(true);
    return true;
}

MyFrame::MyFrame()
    : wxFrame(NULL, wxID_ANY, "Hello wxWidgets", wxDefaultPosition,
wxSize(800, 600)) {}
```

