


Windows

Linux 与 MAC

Windows——Code::Blocks 使用说明

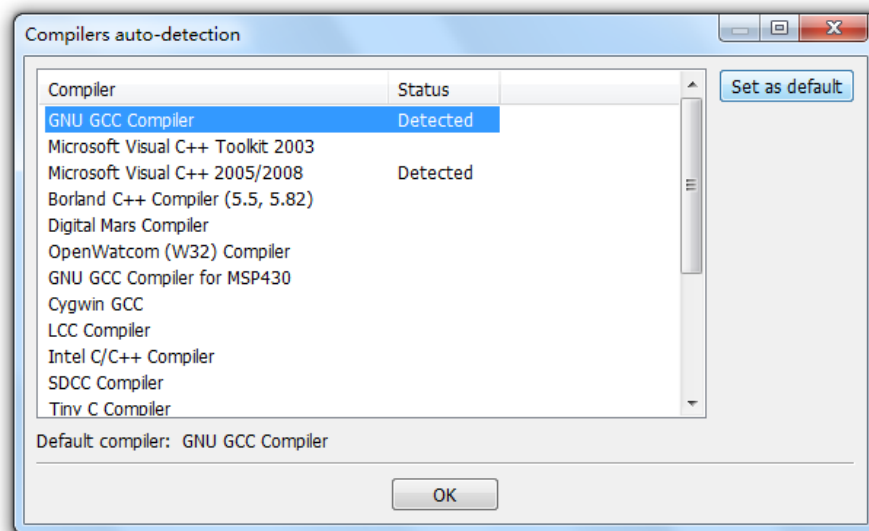
在 <http://www.codeblocks.org/downloads/binaries> 选择合适的平台并下载相应的安装文件。
对于 Windows 平台，需要下载 **codeblocks-17.12mingw-setup.exe** 文件。

 Windows XP / Vista / 7 / 8.x / 10:

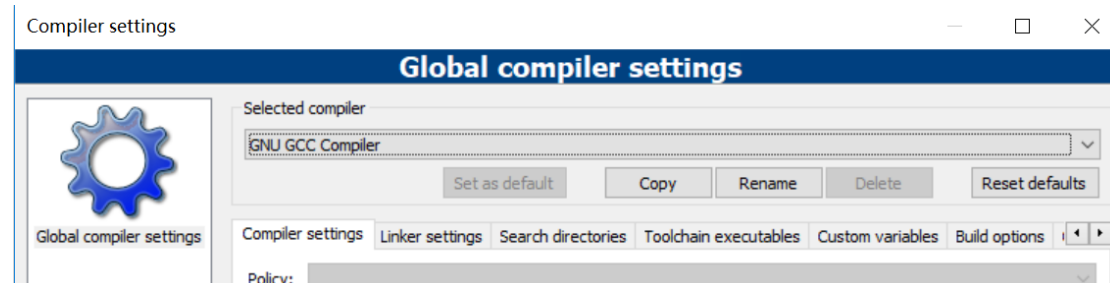
File	Date	Download from
codeblocks-17.12-setup.exe	30 Dec 2017	Sourceforge.net
codeblocks-17.12-setup-nonadmin.exe	30 Dec 2017	Sourceforge.net
codeblocks-17.12-nosetup.zip	30 Dec 2017	Sourceforge.net
codeblocks-17.12mingw-setup.exe	30 Dec 2017	Sourceforge.net
codeblocks-17.12mingw-nosetup.zip	30 Dec 2017	Sourceforge.net
codeblocks-17.12mingw_fortran-setup.exe	30 Dec 2017	Sourceforge.net

下载完成后，进行安装。

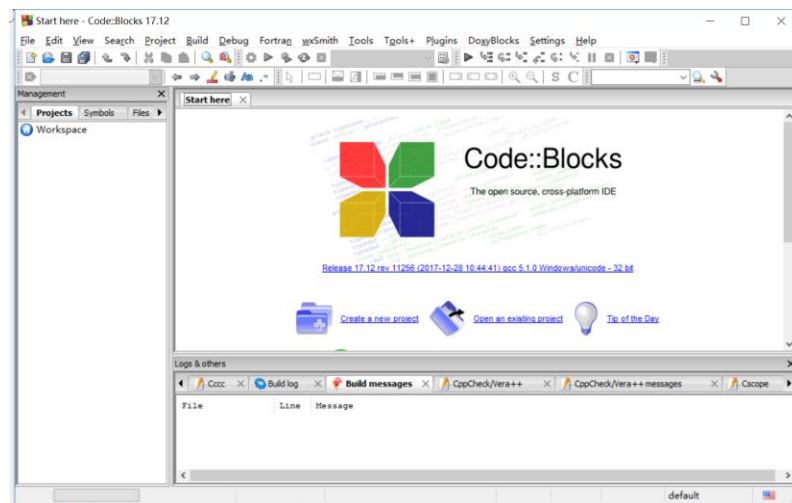
首次打开 Code::Blocks 会出现如下对话框，选择 “GNU GCC Compiler”，再点击右侧的 “Set as default” 选定所需的编译器。



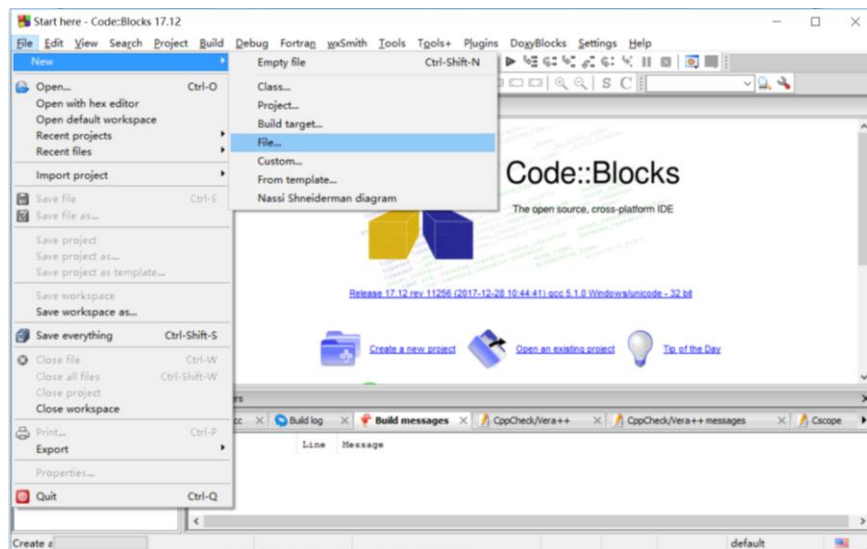
若在程序内需要更改编译器，选择 “Settings” - “Compiler...”，在 “Global compiler settings” 的第一项 “Selected compiler” 中选择 “GNU GCC Compiler” 再点击 “Set as default”。



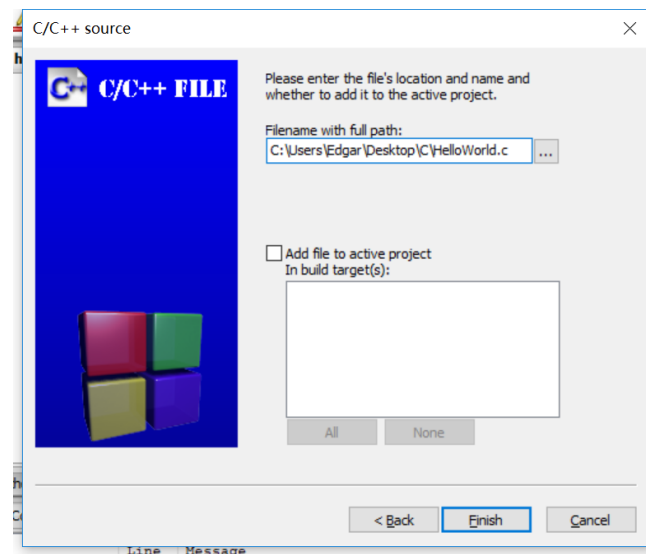
Code::Blocks 启动后能看到如下界面：



若需创建单个.c 文件，我们可以选择“File” - “New” - “File...”（无法进行断点调试）



选择“C/C++ source”，在选择语言时选择“C”。选择路径确定文件名后点“Finish”完成。



下面是简单的 “Hello World” 示例代码：

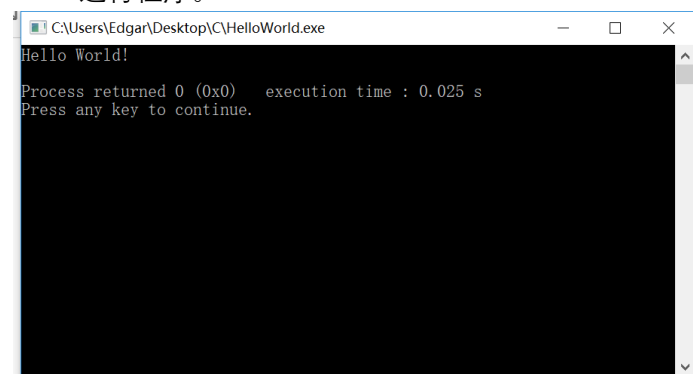


```
1  #include <stdio.h> // Header
2
3  int main() // Main function
4  {
5      printf("Hello World!\n"); // Print "Hello World!"
6      return 0;
7  }
```

完成代码后，需要进行编译和运行。

选择 “Build” - “Build” 进行编译。如果有错误，会显示在下方 “Build Messages” 中，可以根据提示的 Error 或 Warning 来修改代码。

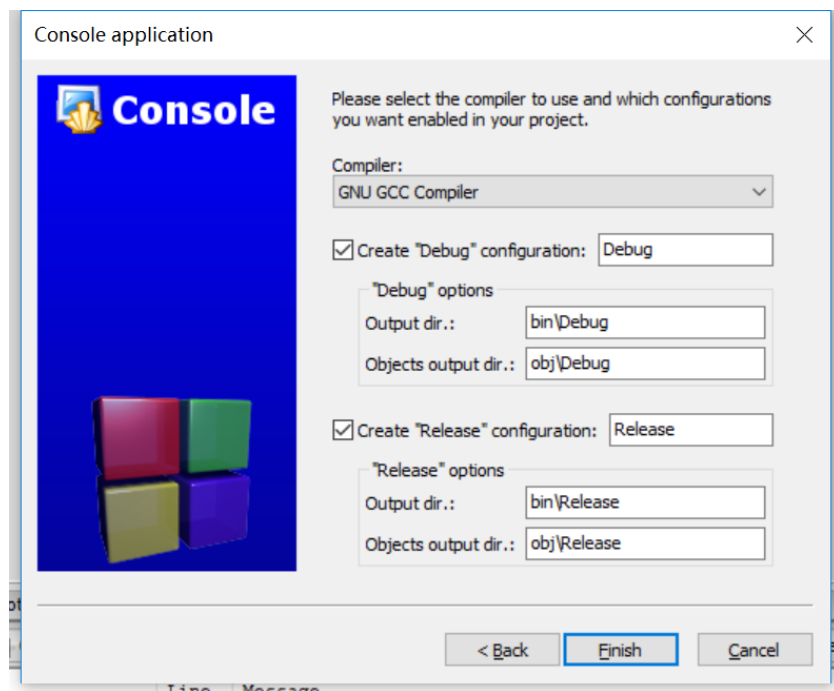
选择 “Build” - “Run” 运行程序。



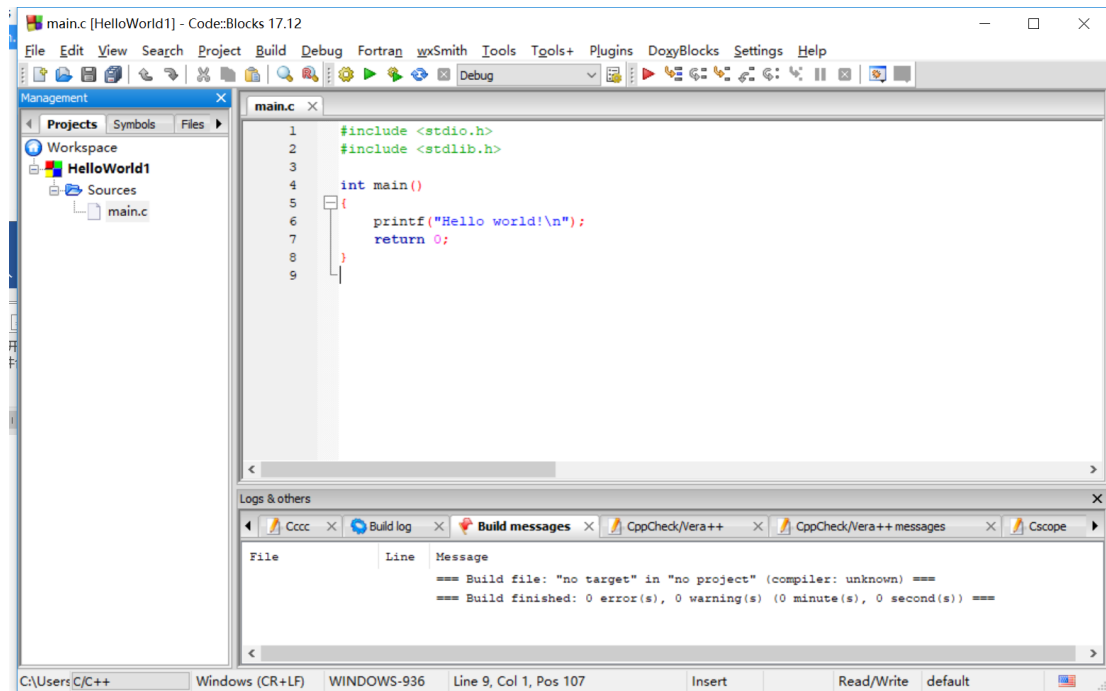
```
C:\Users\Edgar\Desktop\C\HelloWorld.exe
Hello World!
Process returned 0 (0x0)   execution time : 0.025 s
Press any key to continue.
```

如果需要创建一个 Project（项目需要许多 header 和 source 文件构成时），选择 “File” - “New” - “Project...”，选择 “Console application”。

填写完文件名和路径后，选择 “Next”，一般都为默认选择，注意 Compiler 选择 “GNU GCC Compiler”。



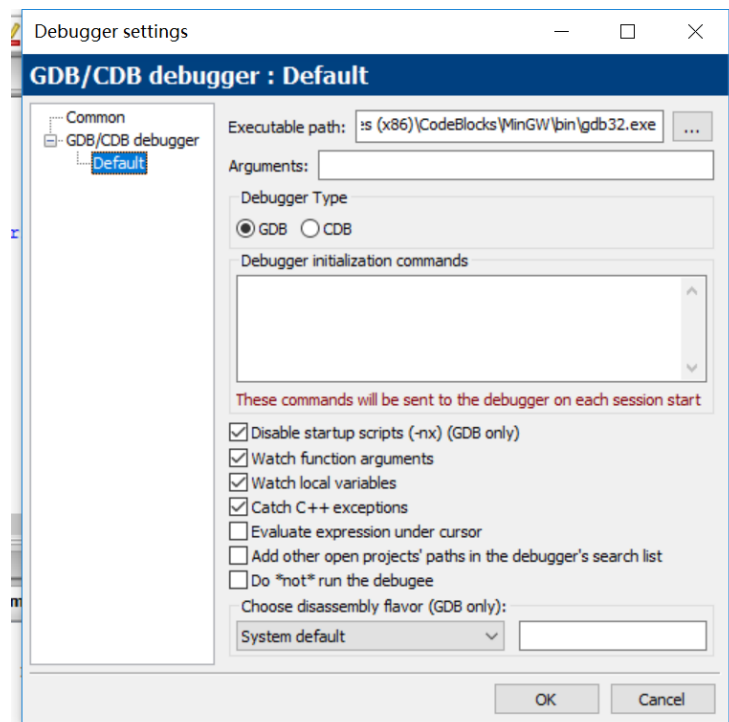
在左侧的 **Projects** 视图中，展开“**Sources**”文件夹，双击“**main.c**”，在右侧的编辑器视图中就能够看到它的代码，也能在这里进行编辑。



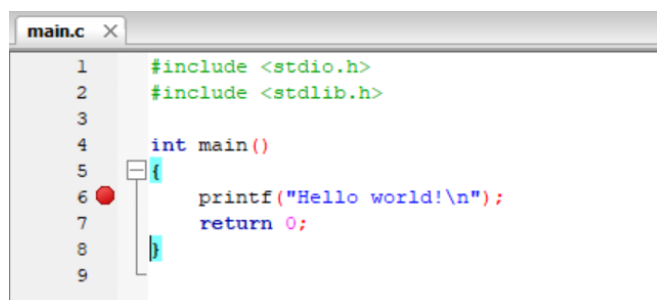
编译与运行操作与之前一样。

在 **project** 中，我们能对程序进行断点调试（**Debug**）。

首先，选择 “**Settings**” - “**Debugger...**” - “**GDB/CDB debugger**” - “**Default**”，设定调试程序路径。



完成后，可以在程序代码中设置断点，在 debug 时就会在断点处暂停等待：



我们可以进行不同操作进行程序调试 。在 "Debugging Window" - "Watches" 可以查看变量状态。

Linux 与 MAC

Linux

在 Linux 系统中，可以选择各种集成环境的软件，如之前提到的 Code::Blocks 等。但因为 Linux 系统中已经有 GCC 编译器，所以我们可以选择在 Terminal 中直接使用命令来进行编译。

首先我们可以通过 gedit、vim 等编辑器来写代码。随后在 Terminal 中定位到当前路径，并进行编译：\$ gcc -o [执行文件] [需要编译的文件]。

```
guest@Edgar:~$ cd ~/C_Programming/
guest@Edgar:~/C_Programming$ gcc -o hello HelloWorld.c
```

随后执行文件：

```
guest@Edgar:~/C_Programming$ ./hello
Hello World!
guest@Edgar:~/C_Programming$
```

MAC

在 MAC 中，我们同样可以在 terminal 中直接对文件进行编译，操作方法与在 Linux 系统中一样。

MAC 中还能够使用 XCode 进行编译（我没有用过 MAC...）。

可以参考 https://blog.csdn.net/cheng_changtui/article/details/79493349。