

VSCode 便携版(集成C/C++编译器)

VSCode Portable WithMinGWw64

本文更新于2018.11.14

随意转载,注明出处 <https://github.com/DIOLeo/VSCode-Portable-WithMinGWw64>

如果您对此文档有什么意见或建议欢迎提出

联系方式 diodev@outlook.com

Feature 特点

- 解压即用,无需配置环境变量
- 集成 MinGW-w64 v6.0.0 (VSCode版本 V1.29.0)
- 集成 cpptools 0.20.1
- 集成 vscode-language-pack-zh-hans 1.29.2
- 可右键以VSCode打开
- 可自由更新

Install 安装方法

1. 下载压缩包
链接: https://pan.baidu.com/s/1-c_53Yi3zw7l40bY0hJ_iw
提取码: ac25
2. 解压缩(非中文目录)
3. 运行安装目录下Start.bat

不要直接运行 Code.exe !!!

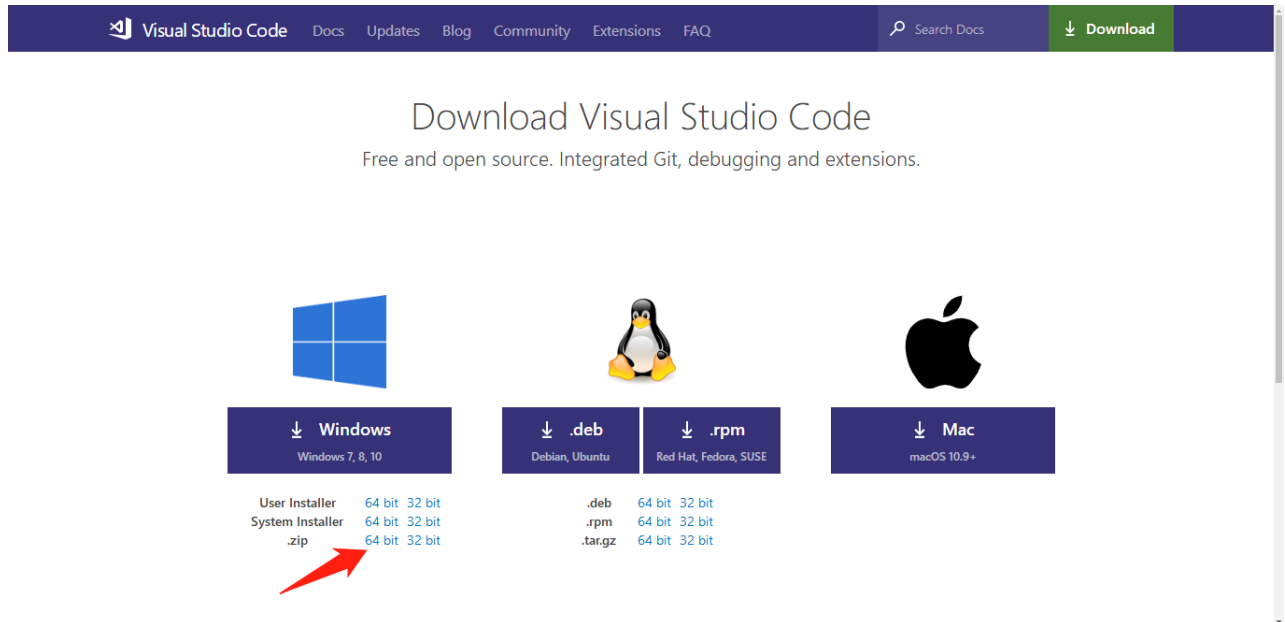
History 更新日志

- V1.0 [2018.11.14]

Document 制作教程

教程略繁琐,需要一定动手能力和足够的耐心和细心,脾气不好请直接下载傻瓜版本
需要的文件:VSCode, MinGW-w64编译环境

1. 下载VSCode压缩包 <https://code.visualstudio.com/Download>



解压到非中文目录, 如 D:\

2. 在安装目录中创建名为 `.portable` 的文件夹

在 `.portable` 文件夹下创建 `User` , `mingw64` 文件夹

在 `User` 文件夹下创建 `.vscode` , `AppData` , `Desktop` 文件夹

在 `.vscode` 文件夹下创建 `extensions` 文件夹

在 `AppData` 文件夹下创建 `Local` , `Roaming` 文件夹

再强调一遍:耐心和细心

3. 回到安装目录, 在安装目录中新建一个TXT文件夹, 将下列代码复制并保存, 更改文件名后缀为`.dat`

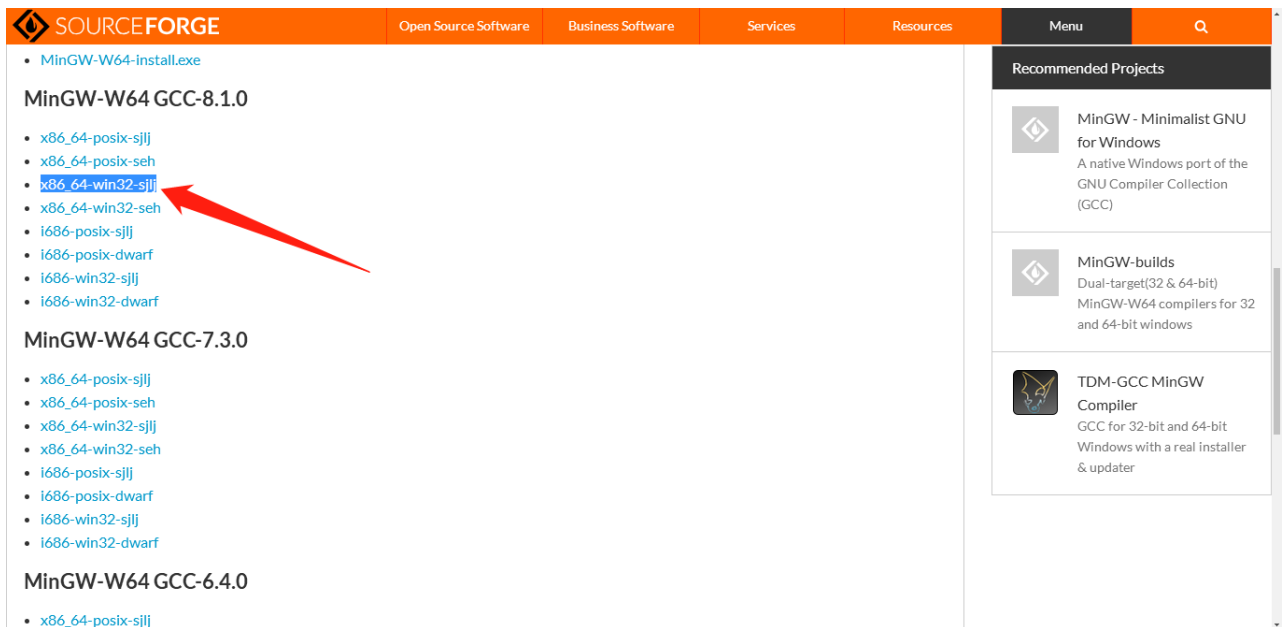
```
@echo off
cd /d %~dp0
call:set_absolute_path USERPROFILE .\.portable\User
call:set_absolute_path APPDATA .\.portable\User\AppData\Roaming
call:set_absolute_path mingwbin .\.portable\mingw64\bin
set path=%mingwbin%;%path%
start Code.exe %1
exit

:set_absolute_path
for /f %%p in ("%2") do (set %1=%~fp)
goto:eof
```

4. 下载编译环境离线安装包 `x86_64-8.1.0-release-win32-sjlj-rt_v6-rev0`

<https://sourceforge.net/projects/mingw-w64/files/>

往下拉, 找到 `MinGW-W64 GCC-8.1.0 -> x86_64-win32-sjlj`



seh结尾是纯64位编译. sjlj结尾是32, 64两种编译, 按需下载

5. 下载完成后解压到刚才创建的 `.portable\mingw64` 目录下
6. 运行安装目录下 `Start.bat`
7. 添加右键菜单 (可选)

右键 `AddShellLink.reg` 用编辑文件打开, 更改其中路径值

双击安装目录下的 `AddShellLink.reg` 合并注册表

附注册表代码

```
Windows Registry Editor Version 5.00

;Open File

[HKEY_CLASSES_ROOT\*\shell\VSCode]
@="Open with Code"
"Icon"="D:\\VSCode\\VSCode-x64\\Code.exe"

[HKEY_CLASSES_ROOT\*\shell\VSCode\command]
@="\"D:\\VSCode\\VSCode-x64\\Start.bat\" \"%1\""
```

```
;Open Folder

[HKEY_CLASSES_ROOT\Directory\shell\VSCode]
@="Open with Code"
"Icon"="D:\\VSCode\\VSCode-x64\\Code.exe"

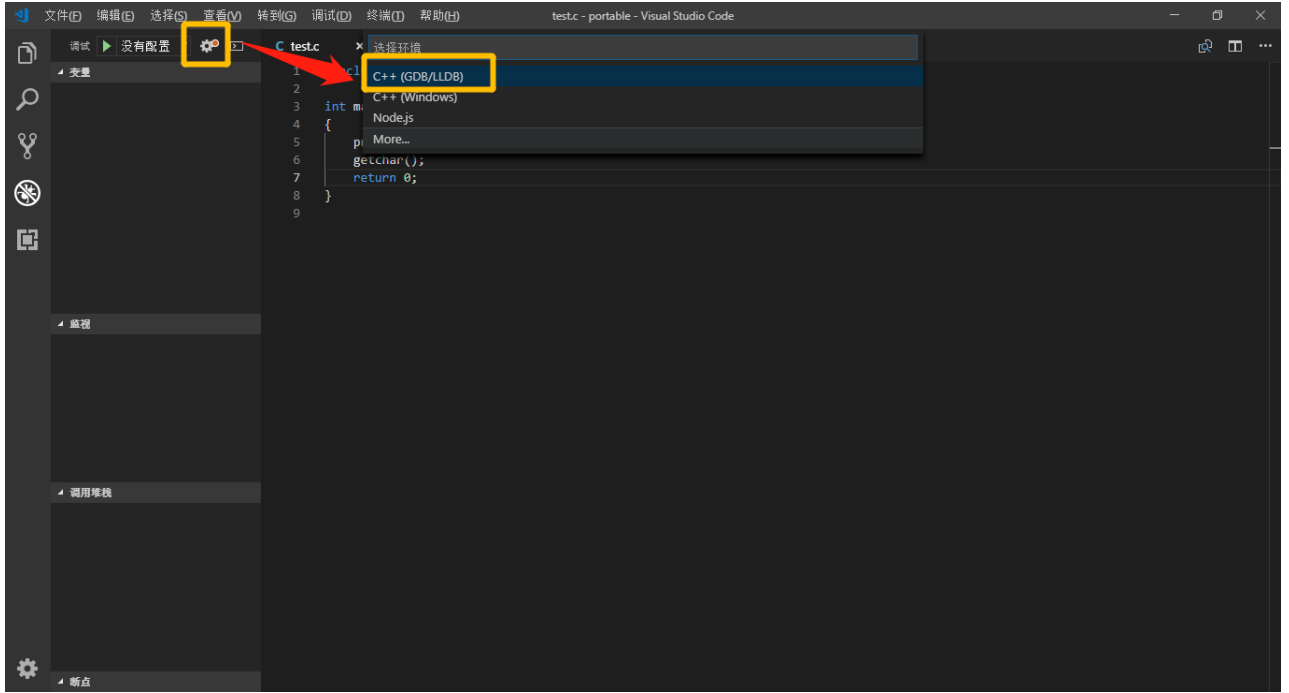
[HKEY_CLASSES_ROOT\Directory\shell\VSCode\command]
@="\"D:\\VSCode\\VSCode-x64\\Start.bat\" \"%V\""
```

Hello World 测试

1. 打开VSCode, 点击左侧文件标志, 打开文件夹, (!注意, VSCode只支持文件夹下构建程序, 从VC6.0过渡过来的需要适应一下!)
2. 新建 `test.c` 文件, 键入 `hello world` 代码
(!注意, VSCode跟某些IDE不同, 生成的exe不会出现"Press any key to continue", 需要手动加

getchar() 或 system("pause")!)

3. 点击左侧调试按钮, 如下图, 添加调试, 随便选一个, 目的是生成 `launch.json` 文件



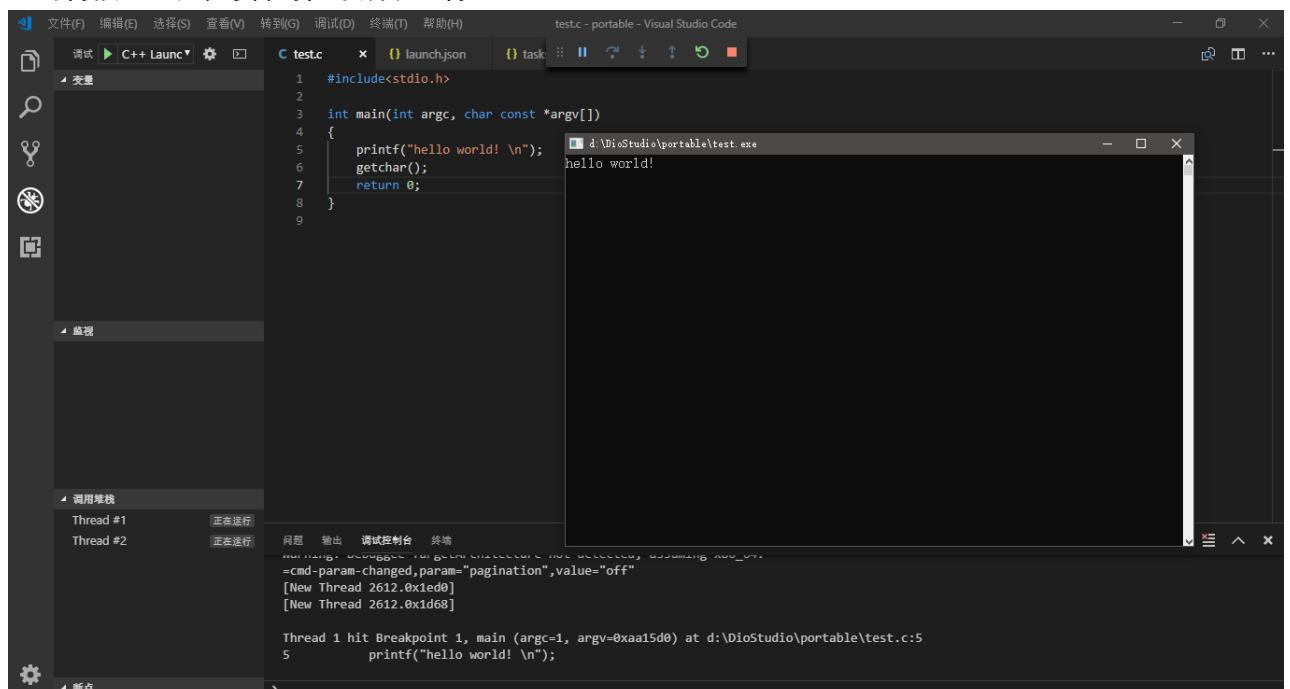
然后将生成的 `launch.json` 文件代码删除, 换成下面的并保存

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "C++ Launch",
      "preLaunchTask": "build",
      "type": "cppdbg",
      "request": "launch",
      "program": "${fileDirname}/${fileBasenameNoExtension}.exe",
      "args": [],
      "stopAtEntry": false,
      "cwd": "${fileDirname}",
      "environment": [],
      "externalConsole": true,
      "MIMode": "gdb",
      "miDebuggerPath": ".\\portable\\mingw64\\bin\\gdb.exe",
      "setupCommands": [
        {
          "description": "Enable pretty-printing for gdb",
          "text": "-enable-pretty-printing",
          "ignoreFailures": true
        }
      ]
    }
  ]
}
```

4. 按F5调试,此时肯定会报错,打开新生成的 `tasks.json` 文件将生成的 `tasks.json` 文件代码删除,换成下面的并保存

```
{
  "version": "2.0.0",
  "tasks": [
    {
      "label": "build",
      "type": "shell",
      "group": {
        "kind": "build",
        "isDefault": true
      },
      "presentation": {
        "echo": true,
        "reveal": "always",
        "focus": false,
        "panel": "shared"
      },
      "windows": {
        "command": "g++",
        "args": [
          "-ggdb",
          "\"${file}\"",
          "--std=c++11",
          "-o",
          "\"${fileDirname}\\${fileBasenameNoExtension}.exe\""
        ]
      }
    }
  ]
}
```

5. 此时再按下F5, 即可看到程序成功运行



6. 大功告成!

Reference 参考文献

VSCode官方文档 <https://code.visualstudio.com/docs>

CSDN_ID:黑猫崽儿 https://blog.csdn.net/c_duoduo/article/details/52083494

DOS命令大全 <https://www.cnblogs.com/zqifa/p/windows-dos-1.html>