



青岛能征智能基础资料

vcpkg安装与配置

修订记录

日期	版本	描述	作者
2025-04-12	1.0.0	初版	王英杰

## 目录

### Windows

- 1.构建vcpkg
- 2.添加环境变量方便vcpkg使用
- 3.visual studio 2019设置
- 4.安装依赖库

### Linux

- 1.安装依赖
- 2.克隆 vcpkg 源码
- 3.构建 vcpkg
- 4.配置环境变量
- 5.安装依赖库
- 7.使用 vcpkg 管理的库
- 8.其他可选配置

# Windows

## 1.构建vcpkg

- 平台 visual studio 2019
- c++库管理工具
- vcpkg + vcpkg 使用方法

```
# 拉取
git clone https://github.com/Microsoft/vcpkg.git
也可以使用 下载好的固定版本的vcpkg 暂定使用 vcpkg-2025.02.14.zip

## 构建
\vcpg\bootstrap-vcpkg.bat

## 集成vcpkg 到visual studio
vcpkg integrate install
```

## 2.添加环境变量方便vcpkg使用

```
# vcpkg 主目录
VCPKG_HOME = git 下载的目录:D:\3_env\2_vcpkg\vcpg 该目录下就是vcpkg可执行程序的目录

# vcpkg 安装编译的默认版本
VCPKG_DEFAULT_TRIPLET = x64-windows

# vcpkg 国内镜像地址, 不设置此项可能需要翻墙才能下载源代码才能编译
X_VCPKG_ASSET_SOURCES = x-azurl,http://106.15.181.5/
```

## 3.visual studio 2019设置

```
工具 -> 选项 -> NuGet包管理器 -> 程序包源 添加-> 名称: vcpkg 源:
{VCPKG_HOME}\scripts\buildsystems 点击更新
```

至此,可以通过vcpkg自动编译所需要的库,并且vcpkg会帮助visual studio 2019自动找编译过程中需要依赖的lib,

visual studio 2019 默认会从{VCPKG\_HOME}\installed\windows\*\include 中寻找头文件,不再需要显式的添加头文

件所在目录.

visual studio 2019 先安装英文语言包,以避免有些库并没有适配中文的项目可以构建导致的库编

译失败.

## 4. 安装依赖库

通过 vcpkg 安装项目需要的依赖库：

```
vcpkg install jsoncpp  
vcpkg install zlib  
vcpkg install libiconv  
vcpkg install spdlog  
vcpkg install polyclipping  
vcpkg install boost
```

# Linux

## 1. 安装依赖

首先，确保你的系统已经安装了以下工具：

- **Git**
- **CMake**
- **G++ (C++ 编译器)**

你可以通过以下命令安装这些工具（适用于 Ubuntu）：

```
sudo apt update  
sudo apt install -y git cmake g++
```

## 2. 克隆 vcpkg 源码

使用 Git 克隆 vcpkg 仓库：

```
git clone https://github.com/microsoft/vcpkg.git
```

也可以直接使用 下载好的固定版本的vcpkg 暂定使用 vcpkg-2025.02.14.zip

### 3.构建 vcpkg

进入 `vcpkg` 目录并运行 `bootstrap` 脚本来构建 vcpkg :

```
cd vcpkg
bootstrap-vcpkg.sh
```

该命令会自动配置系统, 使得 CMake 可以使用 vcpkg 管理的库。

### 4.配置环境变量

为方便在系统中使用 vcpkg, 可以将 vcpkg 的路径添加到环境变量中。在 `~/.bashrc` 文件 (如果使用的是 bash shell) 中添加以下内容 :

```
export VCPKG_ROOT=/path/to/vcpkg
export PATH=$VCPKG_ROOT:$PATH
```

然后执行以下命令使更改生效 :

```
source ~/.bashrc
```

### 5.安装依赖库

现在, 你可以使用 vcpkg 安装所需的库。比如, 要安装

`rapidjson`、`libcurl`、`libiconv`、`cryptopp` 等库, 可以使用以下命令 :

```
vcpkg install rapidjson
vcpkg install libcurl
vcpkg install libiconv
vcpkg install cryptopp
vcpkg install sqlite3
vcpkg install log4cxx
```

如果你希望为不同的架构安装库（如 64 位或 32 位），可以使用 `x64-linux` 或 `x86-linux` 作为 triplet。例如，安装 64 位的 `libcurl`：

```
vcpkg install libcurl:x64-linux
```

## 7.使用 vcpkg 管理的库

在使用 CMake 构建项目时，确保 vcpkg 已经集成，并且在 CMake 配置中能够找到这些库。你可以通过设置 `CMAKE_TOOLCHAIN_FILE` 来告诉 CMake 使用 vcpkg：

```
cmake -DCMAKE_TOOLCHAIN_FILE=/path/to/vcpkg/scripts/buildsystems/vcpkg.cmake ..
```

## 8.其他可选配置

指定国内镜像（如果遇到下载速度慢的问题），可以设置 vcpkg 使用国内镜像下载源：

编辑 `~/.bashrc` 文件，加入如下配置：

```
export X_VCPKG_ASSET_SOURCES="x-azurl,http://106.15.181.5/"
```

然后执行：

```
source ~/.bashrc
```