



VC++ Packaging Tool

#vcpkg #visual-studio #libraries #windows #cpp #package-manager #packages #c #cplusplus #cmake

6,147 commits 42 branches 0 releases 276 contributors

Branch: master New pull request Find file Clone or download

Fetching latest commit...		
docs	Add crossplatform announcement to docs	Apr 24, 2018
ports	Merge pull request #3256 from eao197/master	Apr 26, 2018
scripts	[bootstrap.sh] Minor refactoring for clarity	Apr 26, 2018
toolsrc	Fix #3170 issue with /permissive-	Apr 26, 2018
triplets	Add triplets for x64-linux and x64-osx	Apr 24, 2018
.gitattributes	[everything] Use -text to ensure consistent files across machines.	Feb 27, 2018
.gitignore	Update gitignore to exclude the `vcpkg` binary (artifact of the build	Apr 25, 2018
.vcpkg-root	Add .vcpkg-root file that signifies the root directory	Sep 20, 2016
CHANGELOG.md	Update CHANGELOG and bump version to v0.0.110	Apr 24, 2018
CONTRIBUTING.md	Moved CONTRIBUTING.md to root. Fixes #54.	Sep 21, 2016
LICENSE.txt	Initial commit	Sep 19, 2016
README.md	Update Readme to reflect cross-platform support (#3304)	Apr 26, 2018
bootstrap-vcpkg.bat	Fix bootstrap-vcpkg.bat to work with space in path	Apr 17, 2017
bootstrap-vcpkg.sh	[sh] Tweak so only the sh forwarder needs to be +x	Apr 26, 2018

README.md

Vcpkg build succeeded

Overview

Vcpkg helps you manage C and C++ libraries on Windows, Linux and MacOS. This tool and ecosystem are currently in a preview state; your involvement is vital to its success!

For short description of available commands, run `vcpkg help`.

Quick Start

Prerequisites:

- Windows 10, 8.1, 7, Linux, or MacOS
- Visual Studio 2017 or Visual Studio 2015 Update 3 (on Windows)
- Git
- Optional: CMake 3.10.2*

To get started:

```
> git clone https://github.com/Microsoft/vcpkg
> cd vcpkg

PS> .\bootstrap-vcpkg.bat
Ubuntu:~/ $ ./bootstrap-vcpkg.sh
```

Then, to hook up user-wide integration, run (note: requires admin on first use)

```
PS> .\vcpkg integrate install
Ubuntu:~/ $ ./vcpkg integrate install
```

Install any packages with

```
PS> .\vcpkg install sd12 curl
Ubuntu:~/ $ ./vcpkg install sd12 curl
```

The best way to use installed libraries with CMake is via the toolchain file `scripts\buildsystems\vcpkg.cmake`. To use this file, you simply need to add it onto your CMake command line as `-DCMAKE_TOOLCHAIN_FILE=[vcpkg root]\scripts\buildsystems\vcpkg.cmake`.

In Visual Studio, you can create a New Project (or open an existing one). All installed libraries are immediately ready to be `#include 'd` and used in your project without additional configuration.

For more information, see our [using a package](#) example for the specifics.

Additional notes on macOS and Linux support can be found in the [official announcement](#).

Tab-Completion / Auto-Completion

`vcpkg` supports auto-completion of commands, package names, options etc. To enable tab-completion in Powershell, use

```
.\vcpkg integrate powershell
```

and restart Powershell.

Examples

See the [documentation](#) for specific walkthroughs, including [using a package](#) and [adding a new package](#).

Our docs are now also available online at ReadTheDocs: <https://vcpkg.readthedocs.io/>!

See a 4 minute [video demo](#).

Contributing

Vcpkg is built with your contributions. Here are some ways you can contribute:

- [Submit Issues](#) in vcpkg or existing packages
- [Submit Fixes and New Packages](#)

Please refer to our [Contribution guidelines](#) for more details.

This project has adopted the [Microsoft Open Source Code of Conduct](#). For more information see the [Code of Conduct FAQ](#) or contact opencode@microsoft.com with any additional questions or comments.

License

Code licensed under the [MIT License](#).