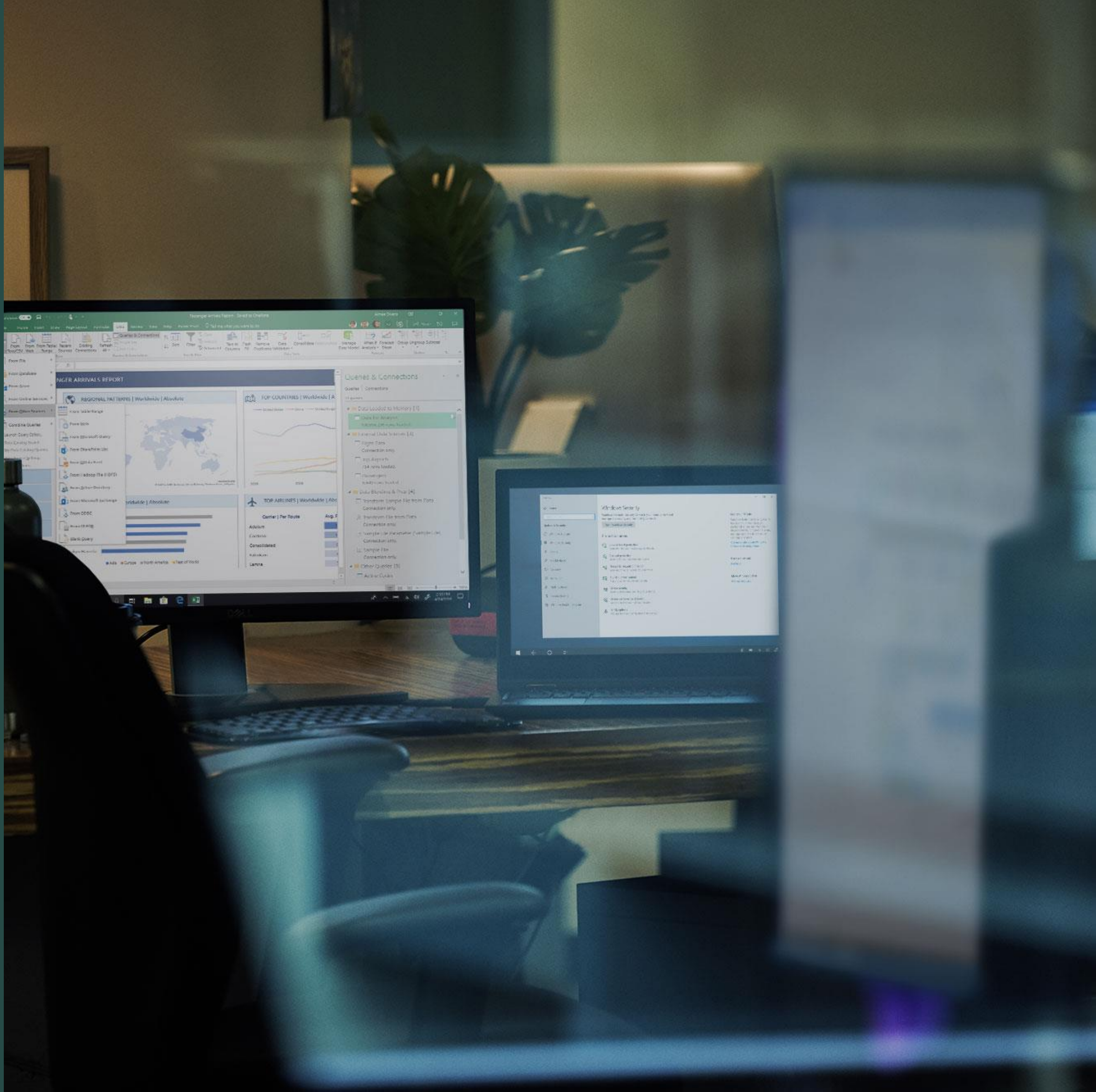
Microsoft

Using vcpkg at work to manage your C++ libraries

Augustin Popa
@augustin_popa

Program Manager
Microsoft C++ Team



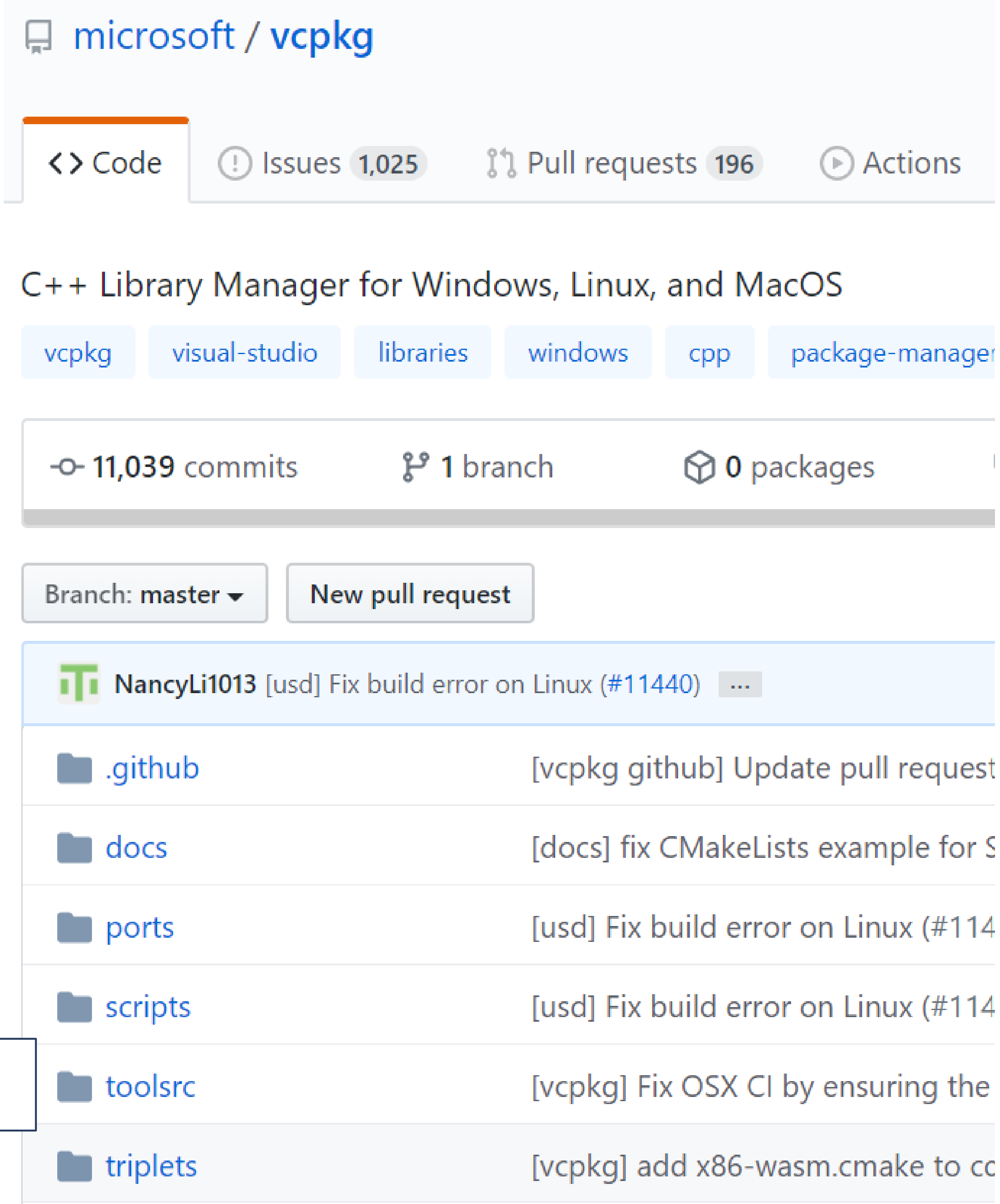
What is vcpkg?

Open source C++ library manager for Windows, Linux, and macOS

1300+ popular open source libraries available as recipes (ports):

- Built from source on-demand
- Centralized, tested catalog

https://github.com/microsoft/vcpkg



microsoft / vcpkg

C++ Library Manager for Windows, Linux, and MacOS

11,039 commits 1 branch 0 packages

NancyLi1013 [usd] Fix build error on Linux (#11440) ...

.github	[vcpkg github] Update pull request
docs	[docs] fix CMakeLists example for S
ports	[usd] Fix build error on Linux (#114
scripts	[usd] Fix build error on Linux (#114
toolsrc	[vcpkg] Fix OSX CI by ensuring the
triplets	[vcpkg] add x86-wasm.cmake to cc

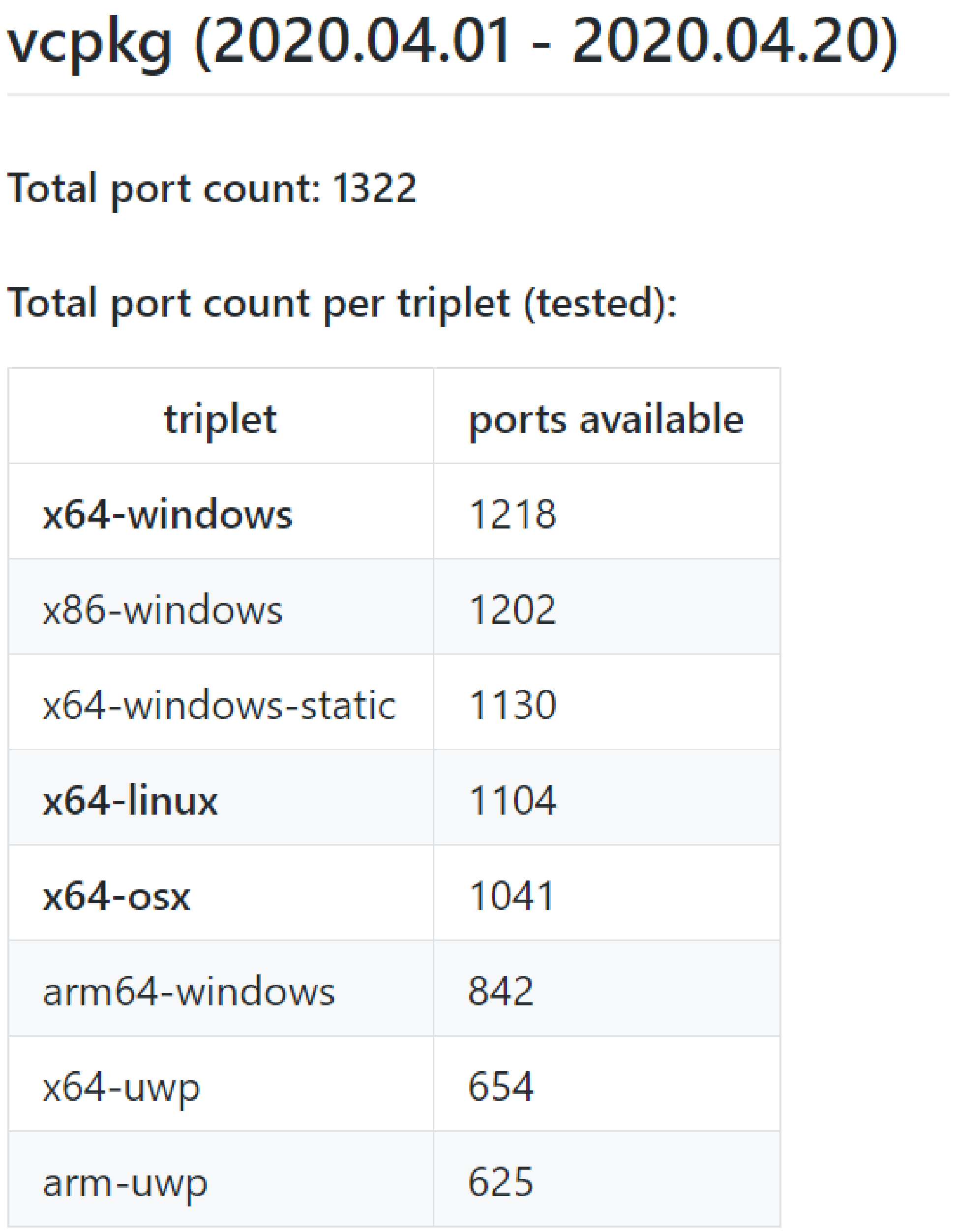
vcpkg catalog count

Terminology:

A **port** is a recipe for building a library

A **triplet** describes the build configuration (target architecture, OS, etc)

The triplets on the right are provided by default – but custom ones can also be defined



vcpkg (2020.04.01 - 2020.04.20)

Total port count: 1322

Total port count per triplet (tested):

triplet	ports available
x64-windows	1218
x86-windows	1202
x64-windows-static	1130
x64-linux	1104
x64-osx	1041
arm64-windows	842
x64-uwp	654
arm-uwp	625

Why vcpkg?

1. Automate the process of building your dependencies to save time
2. No need to worry about dependencies of dependencies – vcpkg will acquire them automatically
3. Regardless of which libraries you install, they will work together – vcpkg routinely builds the entire catalog to test it
4. Provides a simple, repeatable way to acquire dependencies across multiple environments (developer machines, CI, containers)

How to get started

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

2. `cd vcpkg`

3. Run `bootstrap-vcpkg.bat` (Windows) or `bootstrap-vcpkg.sh` (Linux/macOS)

4. (Optional) If using with Visual Studio or Visual Studio Code

   ```
   vcpkg integrate install
   ```

5. `vcpkg install <lib1> <lib2> <lib3>`

Demo

Getting started with vcpkg

Integrating vcpkg with a build system

- MSBuild – run `vcpkg integrate install`
  - Makes vcpkg installed libraries available to MSBuild automatically

- CMake – reference vcpkg CMake toolchain file
  - *[vcpkg-install-path]/vcpkg/scripts/buildsystems/vcpkg.cmake*
  - If you run `vcpkg integrate install` and are using Visual Studio, the toolchain file is referenced automatically for you

Working with triplets – Examples

```
vcpkg install openssl:x64-windows-static
```

Installs static version of OpenSSL for Windows x64 architectures

```
vcpkg install sqlite3:x64-linux-dynamic
     --overlay-triplets=custom-triplets
```

Installs sqlite3 by following a user-defined build recipe located in the custom-triplets subfolder. The triplet file looks like this:

```
# ~/git/custom-triplets/x64-linux-dynamic.cmake
set(VCPKG_TARGET_ARCHITECTURE x64)
set(VCPKG_CRT_LINKAGE dynamic)
set(VCPKG_LIBRARY_LINKAGE dynamic)
set(VCPKG_CMAKE_SYSTEM_NAME Linux)
```

Exporting vcpkg libraries

```
vcpkg export <pkg1> <pkg2> … --[options]
```

Available options:

- --zip
- --7zip
- --nuget
- --raw [uncompressed folder]

Example: `vcpkg export cpprestsdk zlib –nuget`

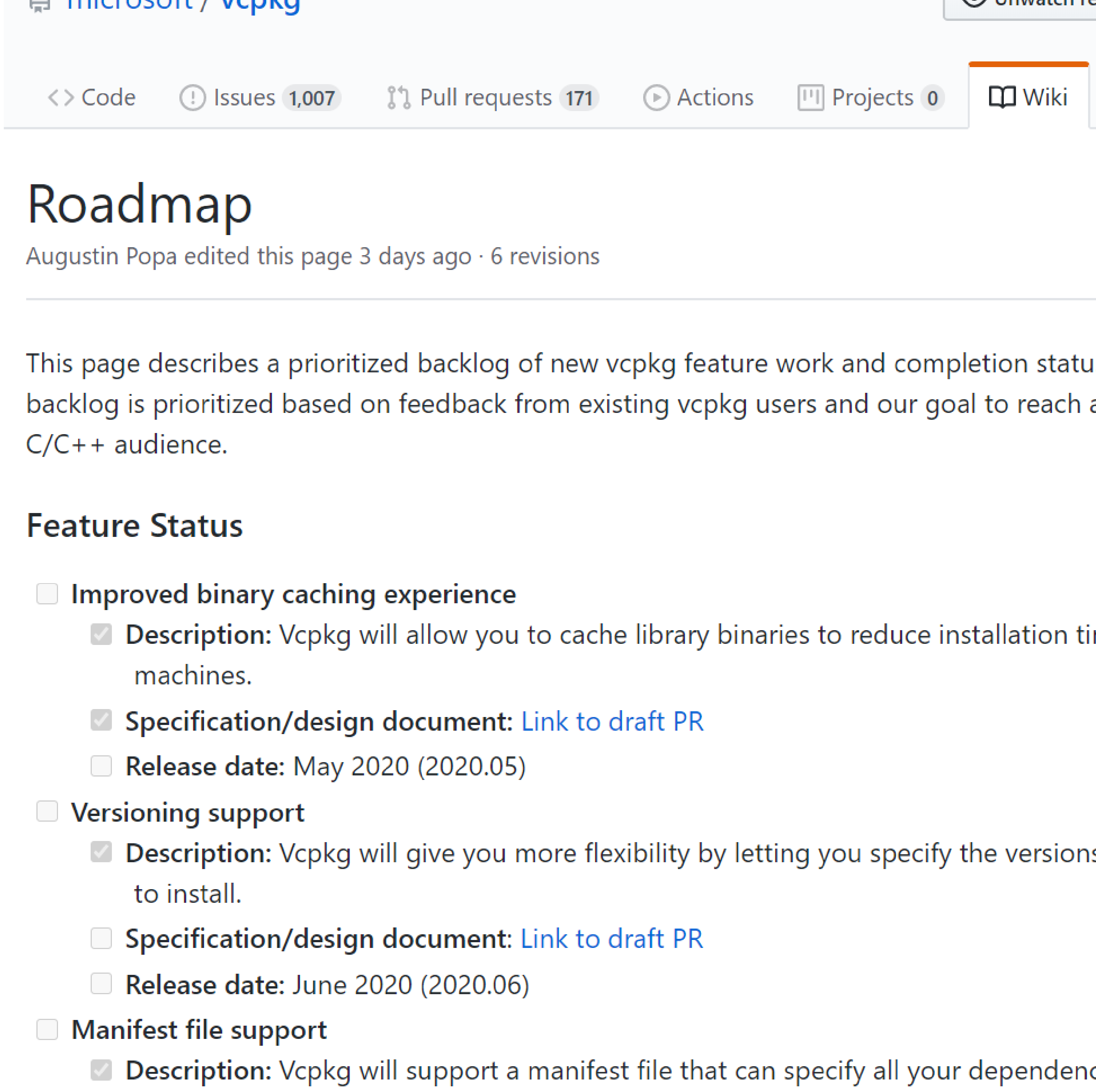
Produces a NuGet package containing cpprestsdk, zlib, and their dependencies that can be used with MSBuild projects/Visual Studio

Coming next to vcpkg…

Product roadmap and feature specifications

https://aka.ms/vcpkg/roadmap

We want your input!


microsoft / vcpkg

Roadmap

Augustin Popa edited this page 3 days ago · 6 revisions

This page describes a prioritized backlog of new vcpkg feature work and completion statu
backlog is prioritized based on feedback from existing vcpkg users and our goal to reach a
C/C++ audience.

Feature Status

- [ ] Improved binary caching experience
  - [x] **Description:** Vcpkg will allow you to cache library binaries to reduce installation ti
    machines.
  - [x] **Specification/design document:** Link to draft PR
  - [ ] **Release date:** May 2020 (2020.05)
- [ ] Versioning support
  - [x] **Description:** Vcpkg will give you more flexibility by letting you specify the versions
    to install.
  - [ ] **Specification/design document:** Link to draft PR
  - [ ] **Release date:** June 2020 (2020.06)
- [ ] Manifest file support
  - [x] **Description:** Vcpkg will support a manifest file that can specify all your dependenc

Binary caching (learn more)

- **The good:** *vcpkg builds from source*, so it can produce tailored, compatible binaries for consumption

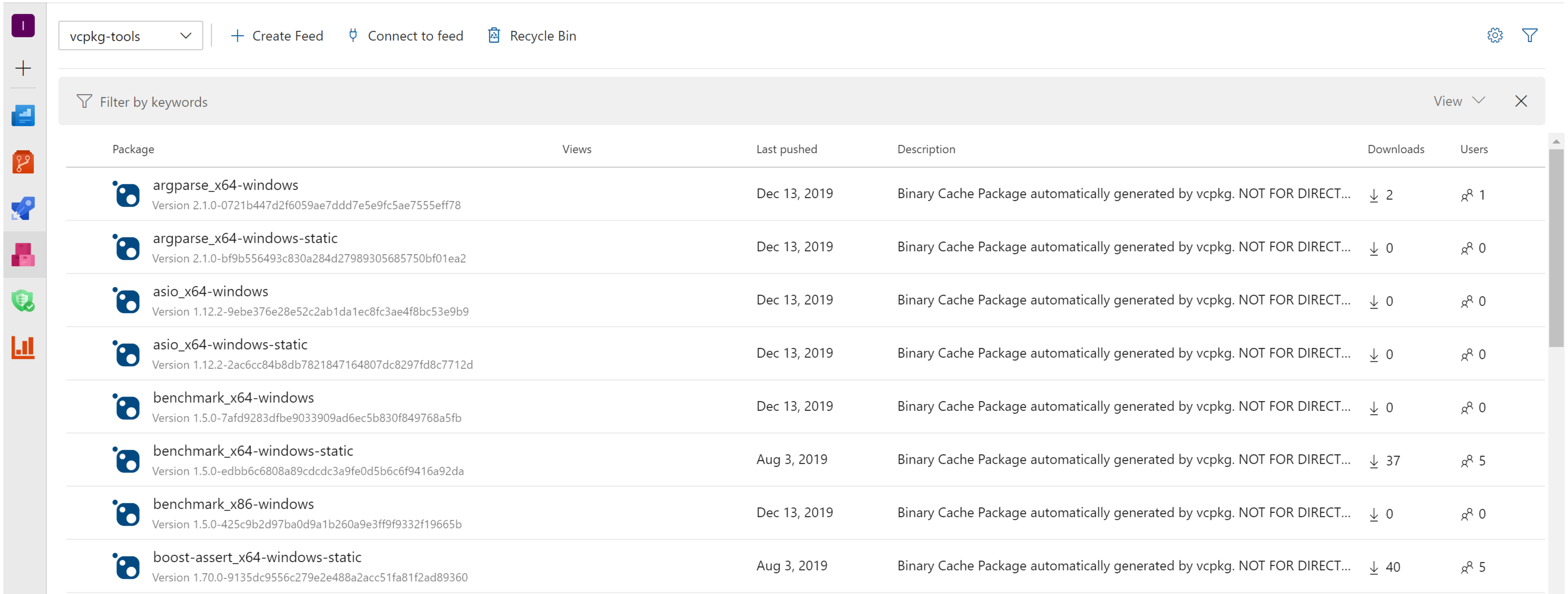
- **The bad:** *vcpkg builds from source*, so it takes a while to install packages for the first time on each machine

- **Solution: Binary caching**
- The first time a library is installed, cache binaries in a known location that can be shared across machines/environments
- Basic example: .zip files in a file-based archive

Binary caching on a NuGet server

Binary caching will also work with existing NuGet servers like Azure Artifact Storage

Note: though storage format is NuGet, packages cannot be consumed directly into MSBuild projects (use `vcpkg export` command instead)



Package	Views	Last pushed	Description	Downloads	Users
argparse_x64-windows Version 2.1.0-0721b447d2f6059ae7ddd7e5e9fc5ae7555eff78		Dec 13, 2019	Binary Cache Package automatically generated by vcpkg. NOT FOR DIRECT...	2	1
argparse_x64-windows-static Version 2.1.0-bf9b556493c830a284d27989305685750bf01ea2		Dec 13, 2019	Binary Cache Package automatically generated by vcpkg. NOT FOR DIRECT...	0	0
asio_x64-windows Version 1.12.2-9ebe376e28e52c2ab1da1ec8fc3ae4f8bc53e9b9		Dec 13, 2019	Binary Cache Package automatically generated by vcpkg. NOT FOR DIRECT...	0	0
asio_x64-windows-static Version 1.12.2-2ac6cc84b8db7821847164807dc8297fd8c7712d		Dec 13, 2019	Binary Cache Package automatically generated by vcpkg. NOT FOR DIRECT...	0	0
benchmark_x64-windows Version 1.5.0-7afd9283dfbe9033909ad6ec5b830f849768a5fb		Dec 13, 2019	Binary Cache Package automatically generated by vcpkg. NOT FOR DIRECT...	0	0
benchmark_x64-windows-static Version 1.5.0-edbb6c6808a89cdcdc3a9fe0d5b6c6f9416a92da		Aug 3, 2019	Binary Cache Package automatically generated by vcpkg. NOT FOR DIRECT...	37	5
benchmark_x86-windows Version 1.5.0-425c9b2d97ba0d9a1b260a9e3ff9f9332f19665b		Dec 13, 2019	Binary Cache Package automatically generated by vcpkg. NOT FOR DIRECT...	0	0
boost-assert_x64-windows-static Version 1.70.0-9135dc9556c279e2e488a2acc51fa81f2ad89360		Aug 3, 2019	Binary Cache Package automatically generated by vcpkg. NOT FOR DIRECT...	40	5

Versioning (learn more)

- **The good:** vcpkg gives you a set of libraries that will work together without the user having to know which versions are compatible

- **The bad:** the user doesn’t easily control the version of a library vcpkg gives them

- **Solution: Versioning support**
- Allows developers to request specific library versions

```
vcpkg install package zlib@1.2.11:x64-windows
```

Package search by version

```
vcpkg search zlib --show-versions

zlib     1.2.11        A compression library
zlib     1.2.10        A compression library
zlib     1.2.8         A compression library
```

Search feature will be able to show available package versions

Manifest file: vcpkg.json (learn more)

- **Problem: How to achieve consistency?**
  - Multiple developers on a team need the same dependencies acquired exactly the same way
  - CI builds need to happen exactly the same way as local developer machine builds
  - Consumers of open source software need to rebuild it the same way as the maintainers
- **Solution:** vcpkg will support a manifest file called vcpkg.json
- Allows developers to specify libraries, library metadata, library versions, and more

vcpkg.json example

```
{
  "name": "pango",
  "version": "1.40.11",
  "port-version": 6,
  "homepage": "https://ftp.gnome.org/pub/GNOME/sources/pango/",
  "description": "Text and font handling library.",
  "dependencies": [
    "glib",
    "gettext",
    "cairo",
    "fontconfig",
    "freetype",
    {
      "name": "harfbuzz",
      "features": [ "glib" ],
      "platform": {
        "and": [
          { "not": { "and": [ "windows", "static" ] } },
          { "not": "osx" } ] } } ]
}
```

Bring your own libraries to vcpkg – package federation

- Eventually, vcpkg.json will allow the user to specify other libraries not found in the vcpkg catalog
- This can include private/internal libraries and custom forks
- Developers will be able to define their own vcpkg ports for use across their organization

Visual Studio / Visual Studio Code integration

- We will ship vcpkg inside the **Visual Studio IDE** (if a C++ workload is installed)
- We will ship vcpkg inside the **Visual Studio Code** C++ extension
- More integration with these tools will be considered over time

Learn more

- vcpkg product roadmap & specs: https://aka.ms/vcpkg/roadmap
  - We are looking for feedback!

- Get started with vcpkg: https://github.com/microsoft/vcpkg