



OpenPrinting

Continuous Integration

Michael Sweet, Lakeside Robotics

September 14, 2022

Continuous Integration



- Mostly Github Actions workflows used to validate/test software changes
- "Build" workflows to build projects on various OS's (Linux, macOS, Windows) and architectures (ARM, x86, etc.)
 - `./configure && make && make test`
- "CodeQL", "Coverity", "LGTm", and other static code analysis workflows to look for errors/issues
- "Docker" for making Docker images
- Also snapcraft, AppImage, Flatpak, etc. builders which are triggered by changes pushed to the corresponding repositories

Github Actions



- Defined in YML files under ".github/workflows" in your repository
- Actions can be triggered by pushes to the repository or started manually
- Private data (access tokens, certificates, etc.) can be added to the environment that is passed
- Some services available under the "Actions" tab on the web site, others you need to setup by hand or through a third-party web site

Example: PAPPPL build.yml



```
name: Build

on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]

jobs:
  build-linux:

    runs-on: ubuntu-20.04

    steps:
    - uses: actions/checkout@v2
    - name: update build environment
      run: sudo apt-get update --fix-missing -y
    - name: install prerequisites
      run: sudo apt-get install -y avahi-daemon
      cppcheck libavahi-client-dev libcups2-dev
      libcupsimage2-dev libgnutls28-dev libjpeg-dev
      libpam-dev libpng-dev libusb-1.0-0-dev zlib1g-
      dev
    - name: configure
      env:
        CC: /usr/bin/gcc
      run: ./configure --enable-debug --enable-
      maintainer
    - name: make
```

```
      run: make
    - name: test
      env:
        ASAN_OPTIONS: leak_check_at_exit=false
      run: make test || cat testsuite/
      testpappl.log
    - name: clang static analyzer
      run: cd pappl && make CC=clang
      "GHA_ERROR:::error::" clang
    - name: cppcheck
      run: cd pappl && make
      "GHA_ERROR:::error::" cppcheck

  build-macos:

    runs-on: macos-latest

    steps:
    - uses: actions/checkout@v2
    - name: install prerequisites
      run: brew install cppcheck libjpeg libpng
      libusb
    - name: configure
      run: ./configure --enable-debug --enable-
      maintainer --enable-sanitizer
    - name: make
      run: make
    - name: test
```

```
      run: make test || cat testsuite/
      testpappl.log
    - name: clang static analyzer
      run: cd pappl && make
      "GHA_ERROR:::error::" clang
    - name: cppcheck
      run: cd pappl && make
      "GHA_ERROR:::error::" cppcheck

  build-windows:

    runs-on: windows-latest

    steps:
    - uses: actions/checkout@v2
    - name: setup-msbuild
      uses: microsoft/setup-msbuild@v1.0.2
    - name: nuget restore
      run: cd vcnet; nuget restore pappl.sln
    - name: msbuild
      run: cd vcnet; msbuild pappl.sln
    - name: test
      run: cd vcnet; ./copy-dlls.bat x64\Debug;
      cd x64\Debug; ./testpappl -c -l testpappl.log
      -L debug -o testpappl.output -t all
```

Output: PAPPPL build.yml



✓ Add length checks to avoid potential pointer wraparound. Build #469

Summary

Jobs

- ✓ build-linux
- ✓ build-macos
- ✓ build-windows

build-linux
succeeded on Jul 19 in 3m 35s

Search logs

- > ✓ Set up job 1s
- > ✓ Run actions/checkout@v2 1s
- > ✓ update build environment 6s
- > ✓ install prerequisites 12s
- > ✓ configure 1s
- > ✓ make 5s
- ▼ ✓ test 1m 17s

```
1 ▶ Run make test || cat testsuite/testpappl.log
6 Testing in doc...
7 Testing in man...
8 Testing in pappl...
9 Testing in testsuite...
10 Good Basic GET: : PASS
11 No Content Length Response: : PASS
12 Bad Basic GET: : PASS
13 Basic HEAD: : PASS
14 Basic POST: : PASS
15 POST Expect w/Continue: : PASS
16 POST Expect w/o Continue: : PASS
17 Good Chunked GET: : PASS
18 Chunked POST: : PASS
19 Bad Chunked GET: : PASS
20 Starting tests...
21 api: papplSystemFindLoc('de'): PASS
22 api: papplLocGetString('A printer with that name already exists.'): PASS (got 'Ein Drucker mit diesem Namen existiert bereits.')
23 api: papplSystemFindLoc('en'): PASS
24 api: papplLocGetString('A printer with that name already exists.'): PASS (got 'A printer with that name already exists.')
```

Printing Projects Using CI



- [CUPS](#): builders and code analysis, runs full CUPS automated test suite
- [ippeveselfcert](#): builders and code analysis, runs self-cert against ippeveprinter
- [ippsample](#): builders and code analysis, runs various IPP standard test files, transforms, and (work in progress) cloud/proxy tests
- [libcups](#) (3.0): builders and code analysis, runs ippeveprinter, ippfind, and ipptool to test basic IPP/2.0 conformance
- [LPrint](#): builders, code analysis, and snapcraft packaging
- [PAPPL](#): builders and code analysis, test suite exercises 90+% of PAPPL API and functionality
- [PDFio](#): builders and code analysis, test suite exercises 100% of PDFio API and functionality

Resources



- AppImage: <https://appimage.org/>
- CodeQL: <https://codeql.github.com/>
- Coverity: <https://scan.coverity.com/>
- Docker: <https://www.docker.com/>
- Flatpak: <https://www.flatpak.org/>
- Github Actions: <https://github.com/features/actions>
- LGTM: <https://lgtm.com/>
- Snapcraft: <https://snapcraft.io/>