NUMFOCUS
PROJECT SUMMIT 2023

Tour of tools used by GDAL CI, testing and documentation

Even Rouault

September 11th 2023

(Very) short introduction to GDAL

- GDAL? Geospatial Data Abstraction Library. The Swiss Army knife for geospatial
- 25 years of history
- C++ code, with Python, Java and CSharp (official) bindings
- Large…
  - ~ 250 raster & vector file formats
  - ~ 70 potential third-party dependencies
  - ~ 1.5 MLoC of C/C++ code for the library
  - ~ 350 kLoC for Python test suite

Developer tools

- CMake build system
- Pre-commit hooks for automated code formatting:
  - flake8, isort, black for Python
  - clang-format for C/C++ code

Testing framework

- GoogleTest for C/C++ code
- pytest for Python code… and C/C++ code (through Python bindings)
- Integrated with CMake’s ctest

Continuous Integration: overview

- 29 checks, mostly on GitHub Actions workflows
- Builders for Linux (Ubuntu, Alpine, Fedora), Windows (native and mingw64) and Mac
- Mix of stable and rolling distributions to get variety of versions of third-party dependencies
- Mix of quick & extensive (slow) builds
- Using Docker for most Linux setups (helps local reproduction)
- Cross-compilation for Android
- Conda-Forge builds (using recipe-clobber to build dev version), and publication of artifacts of master in a dedicated channel
- 2 Travis-CI jobs for s390x (big endian) and ARM64

⇒ Everything MIT licensed: https://github.com/OSGeo/gdal/tree/master/.github/workflows

Continuous Integration: code checks / analysis

Compilers:

- Rather pedantic gcc/clang warning levels (but not -pedantic !), with warning-as-errors (on CI)

Static analyzers:

- CLang Static Analyzer
- Cppcheck
- Coverity Scan (proprietary, free plan for O.S. projects): weekly runs

⇒ useful but significant rate of false positives

Dynamic analyzers:

- Address & undefined sanitizers enabled builds (gcc/clang -fsanitize= flags)

Continuous Integration: code checks / analysis

Custom checks:

- custom scripts to detect some bad practices not detected by compilers (self assignment, missing includes, etc.)
- check-jsonschema for .json resources
- xmllint --validate for .xml resources

Code Coverage:

- For C/C++ code only. Using gcov, lcov
- Integrated with Coveralls.io online service
- Manually publication of lcov HTML in a github repo for nice HTML browsing

Continuous Integration: security testing

- GDAL integrated with OSSFuzz: nightly builds, run on OSSFuzz infrastructure
- CI-Fuzz continuous integration (GitHub Actions workflow)

Documentation

- Integrated in main repository
- Sphinx-based (.rst):
  - Breathe to import Doxygen output for C/C++ code
  - Sphinx automodule for Python API
  - Integration (copying of) javadoc HTML output for Java API