2018 International Radiance Workshop

Automatic for the People - Automated Build/Test/Packaging for Radiance

Rob Guglielmetti

[National Renewable Energy Laboratory - Golden, CO USA]
Outline

Continuous Integration (CI) for Radiance

- Motivation, definition, etc.
- Implementation
  - Workflow
  - GitHub Mirror
  - GitLabCI
- Case Study
- Next Steps
Motivation
Motivation

- OpenStudio project (NREL 2010)
- Cross platform, most users on Windows
- Other efforts:
  - Axel Jacobs (last update ~2016)
  - Francesco Anselmo (mingw, ~2008, no rvu, etc)
  - Georg Mischler (Rayfront ~2002, Windows rvu!)
  - LBNL Desktop Radiance (2000?)

Needed easy way to produce reliable, cross-platform Radiance installers, keep up with HEAD

- Windows versions of GUI apps e.g. rvu
- Releases tied to OpenStudio releases
- CMake build system (implemented ~2013)
How it Works

[GitHub, GitlabCI, CMake]
Workflow

<table>
<thead>
<tr>
<th>CVS (Ward/LBNL)</th>
<th>Git (NREL)</th>
<th>GitLab CI (Google/NREL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>code change</td>
<td>idle</td>
<td>make new release</td>
</tr>
<tr>
<td></td>
<td>refresh mirror (cron)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>new commit(s)?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>no</td>
<td></td>
</tr>
<tr>
<td></td>
<td>push to master</td>
<td></td>
</tr>
<tr>
<td>notify / patch</td>
<td>CI workflow triggered</td>
<td>failure(s)?</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>
GitHub Mirror of CVS
GitHub Releases Page

This is NREL Radiance release "5.2.7", based on the official Radiance 5.2a source code hosted at LBL.

The patch-level increment is simply to keep track of release order, but internally the binaries will report the sha (unique id) of the commit the source code was at when the packages were created. e.g., `nrelrad-5.2.7.tar.gz`

Renamed on this release is:

RADIANCE 5.2.7.tar.gz NREL 2018-06-28 (based on RADIANCE 5.2a Official Release by G. Ward)

The state of the source tree at the time these packages were created can be accessed here.
GitHub Issues Page
GitHub Issues Page (cont'd.)

![GitHub Issues Page](image-url)
GitLabCI

Continuous Integration

- Monitors Git Repo
- Any change triggers new build/test/deploy
GitLabCI - Projects Page (overview)
GitLabCI - Project Detail
The GitLabCI Experience

[and a case study]
Hi Greg,

I have some small changes to evalglare (change of default values) and an extended version of gendaylit (color option included). Both should compile without problems (I tested it and had also only small changes).

If you have time, please include them into the CVS.

Important: gendaylit needs now a second cal file (for the colored option, so please add the perezlum_c.cal to the lib folder).

Both tools have also updated manpages which are also included in the tar files.

Thx!

See you on Monday!

best

Jan
GitLabCI - Build fail
GitLabCI - Build fail - STDOUT
GitLabCI - CVS add
GitLabCI - Build Success

![GitLabCI build success screenshot](image-url)
GitLabCI - Tests Passed, Make new release?
New 'Workshop Edition' Release
Next Steps

More Testing

- Currently the CMake Test suite is a handful of regression, and simple sanity tests
- Need more tests, more coverage
- There is some DOE funding for this in FY19!

Improved Packages

- Still missing some supplemental files
  - `picture.cal` `dirt.cal` (and more!)
- Better control of setting `PATH`, `RAYPATH`, other env vars (Mac)
- Build pdfs of manpages for Windows users
  - Mac/Linux get manpages installed to `[radiance_install_dir]/man` automatically
Next Steps (Cont'd.)

More Package Options

- Better linux package options
  - Current "installer" is just a tarball of 64-bit executables and libs built on Ubuntu
  - Randolph Fritz working on a MacPorts "port" for Radiance
  - Other packages? (rpm? brew? what? WHAT!?)

More Automation

- Automatic Package Posting?
- Iteration builds (i.e. last successful HEAD commit always available)?

Documentation

- Help individual users make their own builds
- User-level repository
Bonus Slide

Your own Private Repo

1. Clone the NREL Git repository:

```
$ git clone https://github.com/NREL/Radiance.git
```

2. Refresh your local copy whenever you want:

```
$ git cvssimport -o master -v -d \
  :pserver:anonymous@radiance-online.org:/home/cvsd/radiance \
-A ../authors.txt -a ray
```

authors.txt = maps CSH logins to GitHub usernames
Bonus Slide 2

What's a test?

execute_process(
   WORKING_DIRECTORY ${test_output_dir}
   COMMAND evalglare${CMAKE_EXECUTABLE_SUFFIX} -L 400 500 0.6 1.2 -B 0.4 \ 
   -b 2500 -d -c outputimage.hdr ${resources_dir}/evalglare/testimage.hdr
   OUTPUT_FILE evalglare_out.txt
   RESULT_VARIABLE res
)

file(READ ${test_output_dir}/evalglare_out.txt test_output)
if(test_output MATCHES "band:band_omega,band_av_lum,band_median_lum, \ 
   band_std_lum,band_perc_75,band_perc_95,band_lum_min,band_lum_max: \ 
   2.423399 2353.496003 1339.703125 2218.441248 3993.937463 5750.375098 \ 
   68.348633 37142.499489")
   message(STATUS "passed")
else()
   message(STATUS "failed")
endif()
Questions?

robert.guglielmetti@nrel.gov

www.nrel.gov/buildings

github.com/NREL/Radiance

rob.guglielmetti@gmail.com

www.luminousbits.com

[thank you.!]
Questions?

robert.guglielmetti@nrel.gov

www.nrel.gov/buildings

github.com/NREL/Radiance

rob.guglielmetti@gmail.com

www.luminousbits.com

[thank you.!]