Status of the ADVANCE CI/CD

David Brown, Ramon Arcilla, Rebecca Coles, Gustavo Nobre
National Nuclear Data Center

23 February 2023
NCSP TPR, Albuquerque, NM
ADVANCE is the ENDF continuous integration system

- Stood up as quick-n-dirty hack to get ENDF/B-VII.1 out the door
  
  … in 2011

- Faithfully (if at times erratically) serving CSEWG since

- All ENDF files ran through variety of physics & format checks as well as processing codes

---

https://xkcd.com/2730

**How to ensure your code is never reused**

IT TOOK SOME EXTRA WORK TO BUILD, BUT NOW WE’LL BE ABLE TO USE IT FOR ALL OUR FUTURE PROJECTS.

**How to ensure your code lives forever**

LET’S NOT OVERTHINK IT; IF THIS CODE IS STILL IN USE THAT FAR IN THE FUTURE, WE’LL HAVE BIGGER PROBLEMS.

---

Brookhaven National Laboratory
ABSTRACT
In this short paper, we describe the production data approach to data curation. We argue that by treating data in a similar fashion to how we build production software, that data will be more readily accessible and available for broad re-use. We should be treating data as an ongoing process. This includes considering third-party contributions; planning for cyclical releases; bug fixes, tracking, and versioning; and issuing licensing and citation information with each release.

Categories and Subject Descriptors
E.5.3 [Data]: Files - Organization, Structure; E.4.3 [Data]: Coding and information theory - Formal models of communication; H.1 [Information Systems] - Models and principles

General Terms
Management, Documentation, Design, Verification

Keywords
Best practices, Cyclical development and release, Production data
Phase I testing automated for nearly 10 years

<table>
<thead>
<tr>
<th>Code</th>
<th>Test</th>
<th>pre-VII</th>
<th>Now</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAN, STANEF, CHECKR, fudge</td>
<td>File summary complete &amp; correct</td>
<td>🤔</td>
<td></td>
</tr>
<tr>
<td>FIZCON, fudge</td>
<td>ENDF format compliance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIZCON, PSYCHE, fudge</td>
<td>Mathematical correctness (e.g. probabilities valid, covariances positive)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTER, fudge (inter.py)</td>
<td>Physical correctness (e.g. Q, thresholds, energy deposition/KERMA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fudge</td>
<td>Compute &amp; check integral metrics (e.g. RI, thermal cross sections, MACS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADVANCE</td>
<td>Completeness (all outgoing particles, including gammas)</td>
<td>🤔</td>
<td></td>
</tr>
<tr>
<td>ADVANCE</td>
<td>Comparisons to microscopic experimental data (EXFOR)</td>
<td>🤔</td>
<td></td>
</tr>
<tr>
<td>ADVANCE</td>
<td>Assessment of application suitability (e.g. usable for fast reactors or spaceflight)</td>
<td>🤔</td>
<td></td>
</tr>
<tr>
<td>fudge (grokres.py)</td>
<td>Reasonable (e.g. covariances, angular distributions)</td>
<td>🤔</td>
<td></td>
</tr>
<tr>
<td>PREPRO, fudge, NJOY (not SCALE yet)</td>
<td>Can process for user codes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Is state of the art? Is best we can do?</td>
<td>🤔</td>
<td></td>
</tr>
</tbody>
</table>
Major reworking of ADVANCE innards

- Gitlab’s Ci/CD configuration controlled by per-project YAML file anyone who is part of the project can edit.
- To prevent accidental (or intentional?) troubles, must execute checking codes in container
- Gitlab, the Kubernetes master node (development2) and the Kubernetes worker node (ADVANCE2 server) are all behind the BNL FireWall
- Reverse proxy server (outside the BNL FireWall) secures access from the Internet to GitLab.
- Two configuration requirements Gitlab.com had not anticipated:
  - Our cluster must live behind firewall
  - Legacy checking codes need temp space
- Gitlab and BNL/ITD cyber security engineers worked with us to resolve issues.
Simpler reports

- Reports will be per-commit, on any git branch (but probably restricted to review branches for now)
- They must be light weight, but not compromise content
- Solution:
  - Summary markdown, per code, replacing website
  - Any important build artifacts made by code (xsdir, ace files, …)
  - Pictures & lists of bugs in summary markdown

Thanks to:
- Ramon Arcilla (BNL-NNDC) for fighting the fight with GitLab and
- Rebecca Coles (BNL-NNP) for creating the simplified reports
Executes on every commit on every branch

It passed!

Multiple ways to find build artifacts in addition to main project page
Execution controlled by a YAML file

```yaml
variables:
  OS_IMAGE: alpine:3.15.4
  DOCKER_IMAGE: docker:19.03.12

# Set DOCKER_TLS_CERTDIR to "" to disable TLS use
# Otherwise, you get an error of 'client HTTP request to HTTPS server'
  DOCKER_TLS_CERTDIR: ""
  SHARED_PATH: /builds/${CI_PROJECT_PATH}/shared

stages:
- login
- verify

registry_login:
  stage: login
  image:
    name: "${DOCKER_IMAGE}"
  services:
    - name: docker:19.03.12-dind
      alias: docker
      # THIS IS IMPORTANT!
      command: ["--tls=false"]
  script:
    - export DOCKER_HOST=tcp://docker:2375
    - echo $DOCKER_HOST
    - docker login -u "$CI_REGISTRY_USER" -p "$CI_REGISTRY_PASSWORD" git.nndc.bnl.gov:5050
    only:
      changes:
      - "*.endf"
      except:
        changes:
```


An unpacked artifacts.zip file

Each changed evaluation gets its own directory

Full build artifact for each code in own directory

stdout stream from code

Helpful instructions
Sample directory

Open this. “Full_output_…”
Output is in markdown and can be viewed in many text editors and web browsers.

atomi.io does this!
Where are we now

• Final testing of full system on alphas sub library (it is small)
• Will run only on review branches for now
• Once live, will adjust review forms to point to the correct artifact URL

• We hope (expect?) you all will have many comments about the build reports
  • Good comments: dbrown@bnl.gov
  • Bad comments: gnobre@bnl.gov