



Accelerated Climate Modeling
for Energy



ACME's testing process

Jeff Johnson, Jim Foucar
Software Engineering Team
ACME All-Hands Meeting
May 5, 2015

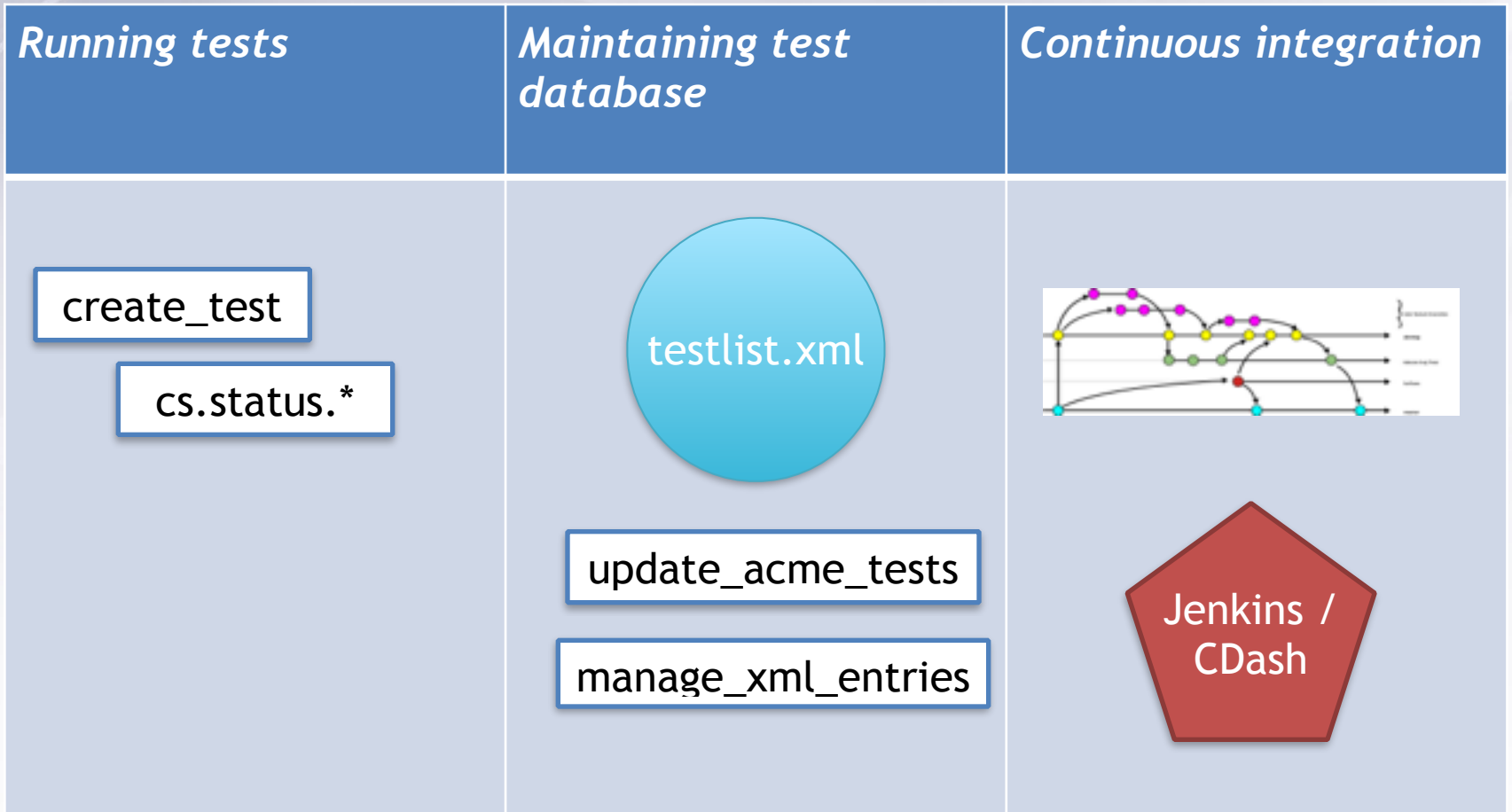
What do we mean by “testing?”

- A test either PASSES or FAILS
 - Establishing criteria is challenging, but worthwhile
 - Result is interpretable by non-experts
- A test should be easy for a non-expert to run
- All important features should be tested
 - Only tested features can be changed reliably!
- It should be easy to run tests frequently
 - Defects are localized in time
- It should be easy for someone to run subsets of tests

Why is testing important?

- Prevents bugs from creeping into the master branch
- Accelerates bug fixes
 - Failing tests provide an “imprint” that gives developers hints about the nature, whereabouts, and timeline of defects
- Builds confidence needed to make transformational software / algorithm / science changes
- Keeps “the model” in a good, reliable (and releasable) state for doing science

ACME's testing machinery



Running a test suite with `create_test`

- Run within ACME/scripts directory
- Many options — not all needed all the time (run without arguments to get help)

```
cd projects/ACME/scripts
mkdir $SCRATCH/acme-baseline
./create_test -xml_mach edison \
              -xml_compiler intel \
              -xml_category acme_developer \
              -testid acme_dev \
              -testroot $SCRATCH/acme_dev \
              -baselineroot $SCRATCH/acme-baseline \
              -project acme
```

Checking the status of a running test suite with `cs.status.*` scripts

- “Test root” directory contains a `cs.status.testid.machine` script

```
cd $SCRATCH/acme_dev
./cs.status.acme_dev.edison
```

...

```
PASS ERS.f19_g16_rx1.A.edison_intel
```

```
PASS ERS.f19_g16_rx1.A.edison_intel.memleak
```

```
PASS ERS.f19_g16_rx1.A.edison_intel.generate./scratch2/
scratchdirs/johnson/acme-baseline-testcases
```

```
FAIL ERS_IOP4c.f19_g16_rx1.A.edison_intel
```

```
BFAIL ERS_IOP4c.f19_g16_rx1.A.edison_intel.generate./scratch2/
scratchdirs/johnson/acme-baseline-testcases
```

```
RUN PEA_P1_M.f45_g37_rx1.A.edison_intel.G.acme_dev
```

```
PEND SMS.ne30_f19_g16_rx1.A.edison_intel
```

...

Interpreting test results

<i>Code</i>	<i>Meaning</i>
<i>PASS</i>	<i>Test passed!</i>
<i>GEN</i>	<i>Test was generated, not run yet</i>
<i>PEND</i>	<i>Test is waiting in a queue</i>
<i>RUN</i>	<i>Test is running</i>
<i>CHECK</i>	<i>Manual review of data required</i>
<i>ERROR</i>	<i>Checking of test result failed</i>
<i>EXPFAIL</i>	<i>Expected test failure</i>
<i>FAIL</i>	<i>Test failed (run failure or inexact restart)</i>
<i>BFAIL</i>	<i>Baseline result could not be found to compare</i>
<i>TFAIL</i>	<i>Test setup error</i>
<i>SFAIL</i>	<i>Generation of test failed in scripts</i>
<i>CFAIL</i>	<i>check_case script failed (env or build problem)</i>

You can add your own tests to existing ACME test suites

1. Edit script to rebuild acme_developer and acme_integration test suites:
`ACME/scripts/acme/update_acme_tests`
2. Modify TEST_SUITES near top of file
3. Change to the directory containing the testing database:
`cd ACME/scripts/ccsm_utils/Testlistxml`
4. Execute the script, rebuilding the database:
`../.. /acme/update_acme_tests suite ./testlist.xml`

You can also add a test case to another pre-existing test suite

1. Change to the directory containing the testing database:

```
cd ACME/scripts/ccsm_utils/Testlistxml
```

2. Extract the existing tests in the desired suite:

```
./manage_xml_entries -query -outputlist -machine machine  
-compiler compiler -category suite > my_tests.txt
```

3. Edit the extracted tests in the file (my_tests.txt), adding one line for each new test.

4. Add the tests in the file to those in the database:

```
./manage_xml_entries -addlist -file my_tests.txt  
-category suite
```

5. Replace the database with the new database:

```
mv testlist-date-time.xml testlist.xml
```

You should generate a baseline for a new test running on a machine

```
cd projects/ACME/scripts
./create_test -testname new-test-case.machine.compiler \  
              -testroot $SCRATCH/new-test-case \  
              -generate new-test-baseline-name \  
              -baselineroot acme-baseline-dir \  
              -project account
```

ACME's testing policy is built into our procedure for code changes

1. Make your code changes in a feature branch
2. Run the `acme_developer` test suite on your branch and check the results to make sure the tests pass
3. If necessary, add your own test case(s) to an existing or new ACME test suite and run it to make sure these new tests pass
4. Issue a pull request and assign an integrator, who
 - integrates the branch into 'next'
 - waits for results of `acme_integration` test suite
 - works with you to address issues
 - integrates the branch into master

Nightly test results are available on CDash (http://my.cdash.org/index.php?project=ACME_Climate)

My CDash All Dashboards Plans & Pricing Support Log Out Tuesday, May 05 2015 00:04:15 EDT

ACME_Climate board Calendar Previous Current Next Project Settings

No file changed as of **Wednesday, April 29 2015 - 21:00 EDT**

4 days ago: 10 tests not run on acme_integration_master_intel
 6 days ago: 10 tests not run on acme_integration_master_intel
 6 days ago: 1 test failed on regression_test
 7 days ago: 1 test failed on regression_test
 7 days ago: 1 test failed on regression_test

See full feed

Nightly

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
redsky	acme_integration_master_intel						10	0	21	Apr 30, 2015 - 04:26 EDT
melvin	acme_developer_master_gnu						0	0	15	Apr 29, 2015 - 22:39 EDT
melvin	acme_developer_next_gnu						0	0	15	Apr 30, 2015 - 05:05 EDT
skybridge	acme_integration_next_intel						0	0	31	Apr 30, 2015 - 03:56 EDT

Nightly test results are available on CDash (http://my.cdash.org/index.php?project=ACME_Climate)



The screenshot displays the ACME Climate dashboard on CDash. The top navigation bar includes links for 'My CDash', 'All Dashboards', 'Plans & Pricing', 'Support', and 'Log Out'. The main header features the ACME logo and the project name 'ACME_Climate', along with navigation buttons for 'board', 'Back', 'Calendar', 'Previous', 'Current', and 'Project'. The main content area shows the following information:

- Testing started on 2015-04-30 08:26:23
- Site Name: redsky
- Build Name: acme_integration_master_intel
- Total time: 16h 47m 8s 940ms
- OS Name: Linux
- OS Platform: x86_64
- OS Release: 2.6.32-504.el6.x86_64
- OS Version: #1 SMP Tue Sep 16 01:56:35 EDT 2014
- Compiler Version: unknown

Below this information, it states '10 tests not run.' and provides a table of test results:

Name	Status	Time
ERB.f19_g16.B1850C5.redsky_intel	Not Run	1h 50s 260ms
ERB.f45_g37.B1850C5.redsky_intel	Not Run	38m 3s 660ms
ERH.f45_g37.B1850C5.redsky_intel	Not Run	17m 58s 320ms
ERS.f09_g16.B1850C5.redsky_intel	Not Run	50m 8s 720ms
ERS.f19_f19.FAMIPC5.redsky_intel	Not Run	23m 18s 540ms
ERS.f19_g16.B1850C5.redsky_intel	Not Run	37m 22s 380ms
ERS.f45_g37.B1850C5.redsky_intel	Not Run	21m 43s 770ms
ERS_IOP_Ld3.f19_f19.FAMIPC5.redsky_intel	Not Run	12m 20s 100ms
ERS_Ld3.f19_g16.FC5.redsky_intel	Not Run	7m 240ms
ERS_Ld3.ne30_ne30.FC5.redsky_intel	Not Run	12m 54s 870ms

At the bottom of the table, there is a link to 'Download Table as CSV File'. The footer of the dashboard includes the Kitware logo and the text 'CDashPro 2.3.0 © Kitware'.

Nightly test results are available on CDash (http://my.cdash.org/index.php?project=ACME_Climate)

My CDash All Dashboards Plans & Pricing Support Log Out

ACME_Climate

board Back Calendar Previous Current Project

Test: ERB.f19_g16.B1850C5.redsky_intel (Not Run)
Build: acme_integration_master_intel (redsky) on 2015-04-30 04:26:23

Exit Value 1

Command line create_test

Show Command Line

Display graphs: Select...

Test output

```
Files /projects/ccsm/ccsm_baselines/master/ERB.f19_g16.B1850C5.redsky_intel/CaseDocs/chem_mech.in and /gscratch:
Files /projects/ccsm/ccsm_baselines/master/ERB.f19_g16.B1850C5.redsky_intel/CaseDocs/cpl_modelio.nml and /gscra
In namelist 'setup_nml', 'ice ic' has inequivalent values 'ERB.f19_g16.B1850C5.redsky_intel.C.150416-105529.ref:
NORMALIZED: 'ERB.f19_g16.B1850C5.redsky_intel.ACTION.TESTID.refl.cice.r.0001-01-10-00000.nc' != 'b40_1850_c02:
FAIL: Namelist diff between files /projects/ccsm/ccsm_baselines/master/ERB.f19_g16.B1850C5.redsky_intel/CaseDoc
Files /projects/ccsm/ccsm_baselines/master/ERB.f19_g16.B1850C5.redsky_intel/CaseDocs/lnd_modelio.nml and /gscra
In namelist 'cam_inparm', 'ncdata' has inequivalent values '/projects/ccsm/inputdata/atm/cam/inic/fv/cami-mam3_
NORMALIZED: cam1-mam3 0000-01-01_1.9x2.5_L30_c090306.nc' != 'b40_1850_c02c_76jpf.cam.i.0221-01-01-00000.nc'
In namelist 'cam_inparm', missing variable: 'cam branch file'
FAIL: Namelist diff between files /projects/ccsm/ccsm_baselines/master/ERB.f19_g16.B1850C5.redsky_intel/CaseDoc
In namelist 'init_ts_nml', 'init_ts_option' has inequivalent values 'ccsm_continue' != 'ccsm_hybrid'
NORMALIZED: 'ccsm_continue' != 'ccsm_hybrid'
In namelist 'init_ts_nml', 'init_ts_file_fmt' has inequivalent values 'nc' != 'bin'
NORMALIZED: 'nc' != 'bin'
In namelist 'overflows_nml', 'overflows_restart_type' has inequivalent values 'ccsm_continue' != 'ccsm_hybrid'
NORMALIZED: 'ccsm_continue' != 'ccsm_hybrid'
In namelist 'iage_nml', 'init_iage_option' has inequivalent values 'ccsm_continue' != 'ccsm_hybrid'
NORMALIZED: 'ccsm_continue' != 'ccsm_hybrid'
FAIL: Namelist diff between files /projects/ccsm/ccsm_baselines/master/ERB.f19_g16.B1850C5.redsky_intel/CaseDoc
Files /projects/ccsm/ccsm_baselines/master/ERB.f19_g16.B1850C5.redsky_intel/CaseDocs/atm_modelio.nml and /gscra
Files /projects/ccsm/ccsm_baselines/master/ERB.f19_g16.B1850C5.redsky_intel/CaseDocs/seq_maps.rc and /gscratch1.
Files /projects/ccsm/ccsm_baselines/master/ERB.f19_g16.B1850C5.redsky_intel/CaseDocs/glc_modelio.nml and /gscra
In namelist 'rtm_inparm', 'finidat rtm' has inequivalent values '/projects/ccsm/inputdata/rof/rtm/initdata/rtm'
```

Upcoming enhancements will make testing easier

- Multiple, coexisting test databases (make your own!)
- Simplified tools for maintaining test databases
- A set of shorter-running test cases for some components (better coverage)
- Better (more specific) diagnostic reporting for troubleshooting test results

You can find more information on ACME Confluence

- Documentation: <https://acme-climate.atlassian.net/wiki/display/Docs/Testing>
- Support page: <https://acme-climate.atlassian.net/wiki/display/SE/Help%3A++Testing> (monitored by SE team)
- Your feedback is appreciated!