A Guide To Using Continuous Integration Within The Verification Environment

Presented by: Jason Sprott, Verilab

Other authors:
Gordon McGregor, Nitero
André Winkelman, Verilab
Jenkins & vManager

Two Examples

Jenkins server

Integration Code

e API
vManager

TCL API
vManager CS

HTML
JUnit

www

API

vManager CS

vPlan
Jenkins(CI) Refresher

1. check in
2. indicate change
3. check out & run
4. results
5. notify & feedback

See: DVCON 2012 “30 Minute Project Makeover Using Continuous Integration”, Gray & McGregor
MDV vs CI

CI (Jenkins)
- Code health
- Keep main branch passing
- Release early & often
- Quickly spot and fix issues
- Integrated with SCM
- Automates test execution
- Communicates status
- PASS/FAIL dashboards

MDV (vManager)
- Finding RTL bugs
- Managing complexity
- Integrated with verification language features
- Verification closure
- Integrates with vplan
- Verification specific results
- Specialized analysis tools
Key Requirements

• Same results on Jenkins as standalone sims
  – Common PASS/FAIL parsing
  – If build fails whole regression is a FAIL
  – CSV or JUnit XML missing is a FAIL
• Regression must run in batch mode
• No return until all threads complete
Two Examples

• Demonstrates generic and specific code

• **Generic:** `csv_to_junit_reporter.py`

• **Ex. 1 (vManager) using a e-language API**
  – Specific: initial launch script
  – Specific: e code to control exit and export results

• **Ex 2 (vManager CS) using TCL API**
  – Specific: initial launch script
  – Specific: TCL to control exit and export results
Regression Launch Script

Export HTML Reports

Export Per-Test Results (as CSV)

JUnit Plug-in

HTML Publisher Plug-in

JUnit Plug-in

HTML Publisher Plug-in

Jenkins

Regression Launch Script

Regression Control & Result Output

Convert CSV to JUnit XML (Python)

Export Per-Test Results (as CSV)

Export HTML Reports

vManager

Import to JUnit plug-in

Import to HTML Publisher plug-in

~90 lines of code

batch_run_e.sh

~45 lines of code

~90 lines of code

vlab_post_session.e

~180 lines of code

csv_to_junit_reporter.py

tool specific

1 build/run

2 Return overall PASS/FAIL

3 Import to JUnit plug-in

4 Import to HTML Publisher plug-in
CSV Output from vManager

Slightly clearer in excel:

<table>
<thead>
<tr>
<th>Test Group</th>
<th>Test Name</th>
<th>Seed</th>
<th>SV Seed</th>
<th>Status</th>
<th>CPU Time (ms.)</th>
<th>Log File</th>
<th>First Failure Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vlab.e_tests</td>
<td>singleline_fail</td>
<td>279134982</td>
<td>random</td>
<td>failed</td>
<td>12.17</td>
<td>.../run_1/local_log.log</td>
<td>Please Jenkins, show this error message!</td>
</tr>
<tr>
<td>vlab.e_tests</td>
<td>passing</td>
<td>1216575217</td>
<td>random</td>
<td>passed</td>
<td>13.48</td>
<td>.../run_2/local_log.log</td>
<td></td>
</tr>
<tr>
<td>vlab.e_tests</td>
<td>multiline_fail</td>
<td>259841377</td>
<td>random</td>
<td>failed</td>
<td>12.66</td>
<td>.../run_3/local_log.log</td>
<td>Please Jenkins, show this error message! Let us see more details of the error. What could have gone wrong? ... no idea ...</td>
</tr>
<tr>
<td>vlab.e_tests</td>
<td>sometimes_fail</td>
<td>1266469697</td>
<td>random</td>
<td>failed</td>
<td>13.54</td>
<td>.../run_4/local_log.log</td>
<td>Unlucky, the test did fail!</td>
</tr>
<tr>
<td>vlab.e_tests</td>
<td>sometimes_fail</td>
<td>863826133</td>
<td>random</td>
<td>passed</td>
<td>13.48</td>
<td>.../run_5/local_log.log</td>
<td></td>
</tr>
<tr>
<td>vlab.e_tests</td>
<td>sometimes_fail</td>
<td>511941856</td>
<td>random</td>
<td>passed</td>
<td>12.3</td>
<td>.../run_6/local_log.log</td>
<td></td>
</tr>
<tr>
<td>vlab.e_tests</td>
<td>sometimes_fail</td>
<td>2105296490</td>
<td>random</td>
<td>passed</td>
<td>12.2</td>
<td>.../run_7/local_log.log</td>
<td></td>
</tr>
<tr>
<td>vlab.e_tests</td>
<td>sometimes_fail</td>
<td>628890105</td>
<td>random</td>
<td>passed</td>
<td>11.43</td>
<td>.../run_8/local_log.log</td>
<td></td>
</tr>
</tbody>
</table>
XML Input to Junit Plugin

Output from `csv_to_junit_reporter.py`

```xml
<?xml version='1.0' encoding='UTF-8'?>
<testsuites name="results">
  <testsuite failures="3" id="0" name="vlab.e_tests" passes="5" time=""/>
    <testcase classname="vlab.e_tests" name="singleline_fail" time="12.17">
      <failure message="Please Jenkins, show this error message!" type="Fail">
        Seed: 279134982
        SV Seed: random
        Log File: /path_to_test/e_tests/run_1/local_log.log
        Complete Message:
        Please Jenkins, show this error message!
      </failure>
    </testcase>
    <testcase classname="vlab.e_tests" name="passing" time="13.48"/>
    <testcase classname="vlab.e_tests" name="multiline fail" time="17.66">
      <failure message="Please Jenkins, show this error message!" type="Fail">
        Seed: 259841377
        SV Seed: random
        Log File: /path_to_test/run_3/local_log.log
        Complete Message:
        Please Jenkins, show this error message!
        Let us see more details of the error.
        What could have gone wrong?
        ... no idea ...
      </failure>
    </testcase>
    <testcase classname="vlab.e_tests" name="sometimes_fail" time="13.54">
      <failure message="Unlucky, the test did fail!" type="Fail">
        Seed: 1266469697
        SV Seed: random
        Log File: /path_to_test/run_4/local_log.log
        Complete Message:
        Unlucky, the test did fail!
      </failure>
    </testcase>
    <testcase classname="vlab.e_tests" name="sometimes_fail" time="13.48"/>
    <testcase classname="vlab.e_tests" name="sometimes_fail" time="12.30"/>
    <testcase classname="vlab.e_tests" name="sometimes_fail" time="12.20"/>
    <testcase classname="vlab.e_tests" name="sometimes_fail" time="11.43"/>
  </testsuite>
</testsuites>
```

Custom data in the failure message
Project Status in Jenkins

Project JenkinsIntegration

Project used for paper preparation regarding Jenkins and vManager integration

Test Result Trend

Permalinks
- Last build (#22), 21 days ago
- Last stable build (#16), 29 days ago
- Last successful build (#16), 29 days ago
- Last failed build (#22), 21 days ago
- Last unstable build (#14), 1 month 21 days ago
- Last unsuccessful build (#22), 21 days ago
Results in Jenkins

Test Result
6 failures (+3)

All Failed Tests
- viabe_tests.singleline_fail
  - Error Details
    - Please Jenkins, show this error message!
  - Stack Trace

All Tests
<table>
<thead>
<tr>
<th>Package</th>
<th>Duration</th>
<th>Fail</th>
<th>Skip</th>
<th>Total</th>
<th>(+/-)</th>
</tr>
</thead>
<tbody>
<tr>
<td>viab</td>
<td>1 min 39 sec</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
Getting the Code

https://bitbucket.org/verilab/jenkinsintegration

Location of `csv_to_junit_reporter.py`
Running vManager

From: vmanager/batch_run_e.sh

vm_launch.pl \
  -vsif ../e/e_tests.vsif \
  -batch \
  -command "load scripts/vlab_post_session.e; sys.vlab_post_session_analysis();" \
  -output_mode log_only

python ../scripts/csv_to_junit_reporter.py regression_tests.csv junit.xml
grep PASS regression_result.txt

A bit of messing around for tidy HTML reports
vManager e API

Excerpt from: vmanager/scripts/vlab_post_session.e

vlab_post_session_analysis() is {

    out("Waiting until all tests have been executed");
    vm_util.trace_all_sessions_to_finish();
    out("All tests should be done by now. Starting post session analysis.");

    vlab_vsof = vm_manager.get_all_sessions()[0]; // or load a specific vsof
    vlab_runs = vlab_vsof.get_runs();

    vlab_out_session_result();
    vlab_out_test_results();
}

There’s about 80 lines of supporting code in total
Running vManager CS

From: vmanagercs/batch_run_e.sh

vmanager -cs -profile ${MY_VMANAGER_CS_PROFILE} -exec scripts/run.tcl
python ..-scripts/csv_to_junit_reporter.py regression_tests.csv junit.xml

From: vmanagercs/scripts/run.tcl

launch -wait -load -vsif ../e/e_tests.vsif
csv_export -runs -view vlabCSV(awg) -out regression_tests.csv -overwrite
report_tests -out tests_report -overwrite
Jenkins Project Configuration

Build
- Execute shell
  Command
  
  #!/bin/bash -x
  . /etc/bashrc
  set -e
  cd SimpleDemo/vm_basic
  export MY_REGRESSION_AREA="pwd"/regression
  ./batch_run.sh

  See the list of available environment variables

Add build step

Post-build Actions
- Publish HTML reports
  HTML directory to archive: SimpleDemo/vm_basic/html_report
  Index page: Regression_Report.html
  Report title: HTML Report
  Keep past HTML reports: checked
  Allow missing report: checked

Add

- Publish JUnit test result report
  Test report XML: SimpleDemo/vm_basic/unit.xml
  Error: Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*.xml'. Basedir of the fileset is the workspace root.
  Retain long standard output/error: unchecked

Delete
Summary

• CI and MDV are complementary
• Two Jenkins integration examples shown
  – Different APIs
• There are some integration requirements
  – e.g. Pass/Fail and exit handling are important
• Generic JUnit XML generation code provided
• CI not just about the tools
  – Cultural change is required