QA Improvements with 1.18

EFL Developer Days 2016

Stefan Schmidt
Samsung Open Source Group
stefan@osg.samsung.com
Agenda

• Coverity
• Unit Testing
• EO API Testing
• Exactness
• Coverage
• Miscellaneous
• Summary
Coverity
Coverity

- Enlightenment down to 0 defects
- Evas generic loaders, emotion generic players and Terminology have been down to 0 already
- Rage has 2 open defects
- EFL (after the merge with Elm) has 124 open defects
- A list with defects grouped by areas will be available for the hacking session
Coverity

Analysis Metrics

Version: efl-2016-05-06-1460321

May 06, 2016
Last Analyzed

1,033,095
Lines of Code Analyzed

0.12
Defect Density

Defect changes since previous build dated May 05, 2016

0
Newly detected

0
Eliminated

Defects by status for current build

1,199
Total defects

124
Outstanding

103
Dismissed

972
Fixed
Coverity

Outstanding Defect per Component

- Elementary
- Edje
- Evas
- Other
- Eio
- Eo
- Embryo
- Static 3rd party libs
- Ecore
- Ecore Evas
- Ecore X
- eldbus
- Ecore Con
- Ecore File
- Ecore IMF
- Eet
- Efreet
- Eina
- Eeze
- Elocation
- Ecore IPC
- Ethumb
- Ecore Cocoa
- Ecore Input
- Ecore Audio
- Ecore Wayland
- Evil
- Emotion
- Ecore Fb
- Ephysics

Legend:
- High
- Medium
- Low

Outstanding defects

0 10 20 30 40
Unit Testing
Unit Testing

• Over 740 test cases by now
• [https://build.enlightenment.org/view/Test%20Coverage/](https://build.enlightenment.org/view/Test%20Coverage/)
• Pick one, there is still plenty of work to do
Unit Testing

• Run test suites individually
• make check-build
• src/tests/evas/.libs/evas_suite

• Or run them as a whole bunch
• make -j 10 check
EO API Testing
EO API Testing

• Autotest script written by Daniel Zaoui
• Bringing various pieces together for API testing
• https://git.enlightenment.org/devs/jackdanielz/autotest.git/
• Eaper and Espion needed to steal the EO APIs for testing
• Coverage information (hit or miss) stored in eet format and can be exported as CVS
Exactness
Exactness on Jenkins

• Exactness job on Jenkins still manually
• To many little problems to run it nightly
• Some tests need to be updated for theme changes
• Investigating on failures with offsets around buttons and textblocks
Coverage
Coverage

• Added initial run to make sure we cover all files and have accurate measurements
• Branch coverage integrated
• Elm test suite integrated during merge
• Code coverage measured for the unit test suite
• Code coverage measured for an Exactness run with elementary_test
Coverage - Unit Tests

- Coverage measurements depend on the used configure options as they define the total lines of code and functions
- A run with the default configure options on my local machine results in the following:

<table>
<thead>
<tr>
<th></th>
<th>Hit</th>
<th>Total</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines:</td>
<td>100256</td>
<td>284444</td>
<td>35.2 %</td>
</tr>
<tr>
<td>Functions:</td>
<td>11740</td>
<td>29469</td>
<td>39.8 %</td>
</tr>
<tr>
<td>Branches:</td>
<td>50240</td>
<td>219872</td>
<td>22.8 %</td>
</tr>
</tbody>
</table>
Coverage - Unit Tests

• On Jenkins the result is a bit lower as we have more code paths enabled:

<table>
<thead>
<tr>
<th></th>
<th>Hit</th>
<th>Total</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines:</td>
<td>89620</td>
<td>327359</td>
<td>27.4 %</td>
</tr>
<tr>
<td>Functions:</td>
<td>10658</td>
<td>32700</td>
<td>32.6 %</td>
</tr>
<tr>
<td>Branches:</td>
<td>47811</td>
<td>222761</td>
<td>21.5 %</td>
</tr>
</tbody>
</table>
Coverage - Exactness

- Running Exactness on the elementary_test test cases
- Right now only 81 of 113 tests cases enabled
- Not covering all elementary_test tests either
- Different code pathes compared to unit test suites
- Should grow rapidly when more tests are working/added
- Collecting code coverage for this shows:

<table>
<thead>
<tr>
<th></th>
<th>Hit</th>
<th>Total</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines:</td>
<td>69912</td>
<td>325687</td>
<td>21.5 %</td>
</tr>
<tr>
<td>Functions:</td>
<td>6967</td>
<td>31171</td>
<td>22.4 %</td>
</tr>
<tr>
<td>Branches:</td>
<td>34095</td>
<td>195979</td>
<td>17.4 %</td>
</tr>
</tbody>
</table>
Coverage

• Merge the different Coverage reports together
• Unit tests + Exactness + EO API testing
• Some need to stabilise more until we can do this
• Should help to understand what parts are covered by our overall testing
Miscellaneous
Code Complexity

- Valid combinations for test matrix
- XCB as x11 lib
- Wayland + DRM
- Harfbuzz, liblz4, web-p, xinput2.2, xine, multisense, lua-old, libvlc, xpresent and hyphen
- systemd is problematic as the build slaves use Gentoo
- EFL profiles: dev, debug and release
API Checker

• Run ABI/API checker still manually
• To many little problems to fully automate it
• Jenkins job prepared at least (Should use it for 1.18)
• Problems with header files during compile
• Due to the intended API/ABI changes for beta APIs the return code always indicates problems
Address Sanitizer

- `-fsanitized=address`
- Quick test CFLAG set during EFL compile broke during eolian generation
- Need to check what causes this and see if we can get a full build through
- Running the test suite and `elementary_test` with it might show us some memory bugs
Summary
Summary

• Pick one of the Coverity issues
• Code Coverage around 30% but should grow once Exactness problems are sorted out
• Merged Coverage report once the various testing stabilises
Thank you. Questions?
Attributions

- Fernando Flores: https://www.flickr.com/photos/ferjflores/8697860914
- Michael Hodge: https://www.flickr.com/photos/mhodge/1216047199/in/photostream/