froglogic

Squish Coco – Cross-Platform Code Coverage Tool Chain

The Coco code coverage tool is a complete, cross-platform, cross-compiler tool chain allowing to analyze the test coverage of C, C++, C#, QML and Tcl code.

Coco

Code Coverage Analysis

Executing a test suite against an instrumented application produces data that Squish Coco can analyze. This analysis can be used to ensure that complete test coverage is achieved since it:

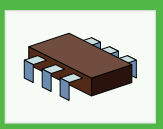
- Finds untested code sections.
- Finds redundant tests (i.e., tests that merely duplicate others).
- Finds dead code (i.e., code that is never executed).
- Computes the optimal order of test execution that will maximize the overall test coverage.
- Compares the test coverage of two applications (e.g., two versions of an application) to identify the differences in test coverage.
- Analysis of test coverage for source code patches in the review process
- Determines minimal set of tests to cover a source code patch

Supported Platforms:


Windows (32- and 64-bit)


Linux (32- and 64-bit)


Mac OS X (32- and 64-bit)


Embedded OSs

UNIX (Solaris, AIX, ...)

In addition, Squish Coco can be made available and supported on other platforms and integrated with custom tool chains.

Learn More and Get in Touch

www.froglogic.com/coco

coco@froglogic.com

Squish Coco can be used at every stage of testing and with every testing method (unit tests, automated tests, manual tests, etc.). Furthermore, Squish Coco can merge multiple execution reports to provide advanced analysis.

Squish Coco is fit to be used to comply with safety standards and regulatory bodies like ISO 26262, EN 50128, DO-178C, IEC 61508 and IEC 62304.

C++ code coverage analysis when the addressbook application is passed the `-h` (help) command line option.

```
1   int main(int argc, char **argv) {
2     QApplication app(argc, argv);
3     if (argc == 2) {
4       QString arg(argv[1]);
5       if (arg == "-h"  ||
6         arg == "--help") {
7         QTextStream out(stderr);
8         return 1;
9         out << "usage: "<<  argv[0] << "\n";
10      }
11    }
12    MainWindow mainWindow;
```

1	Executed code
0-1	argc == 2: was never false
1	Executed code
0-1	arg=="-h": was never false
0	arg=="--help": was never true or false
1	Executed code
1	Executed code
X	Unreachable code
-	
-	
0	Never executed

Result Summary

Lines executed:	4
Lines not executed:	2
Lines of dead code:	1
Expressions not covered:	3
Decision branches not executed:	2

The Squish Coco Tools

CoverageScanner this analyzes and instruments C and C++ applications.

CoverageBrowser a sophisticated GUI tool which displays and manages coverage analysis data and results. (Available to commercial licensees only.)

Microsoft® Visual Studio Add-in this generates code coverage configurations for all available C and C++ projects directly inside the Visual Studio IDE.

Supported Coverage Levels

- Function coverage
- Line and statement coverage
- Decision (or branch) coverage
- Condition coverage
- Condition/decision coverage
- Modified condition/decision coverage (MC/DC)
- Multiple condition coverage (MCC)