Verifysoft TECHNOLOGY

Testwell CTC++ for Java and Android™

Industrial proofed Code Coverage made easy

The quality of your Android™ code is important to you?
Track your Android application code quality with Testwell CTC++ Test Coverage Analyser. Testwell CTC++ presents code coverage metrics for all parts of your Android code. The tool supports all coverage levels and is ready to be used in safety-critical projects.

Advantages of CTC++ for Java and Android™

- Very small instrumentation overhead
- Analyses code coverage on all targets
- Works with even the smallest targets
- Works with any compiler/cross compiler
- No changes in your source code needed
- Supports java compiler (Java part)
- Source code based
- Easy integration in nearly every IDE

Flexibility of usage

- Runs on Windows, OSX and Linux systems
- Can easily be integrated into your projects
- Works with Ant build systems
- SDK, NDK, PDK, Kernel. . .
- With Ant, ndk-build . . .
- Works transparently to the build system
- No shadow directories
- Works on non-rooted phones/tablets
- Works on release devices

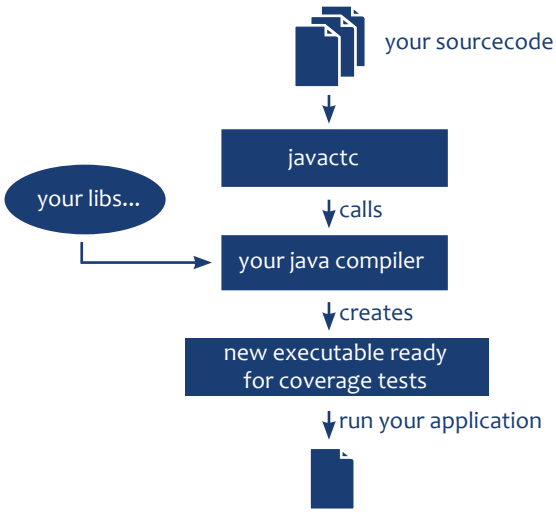
Verifysoft TECHNOLOGY

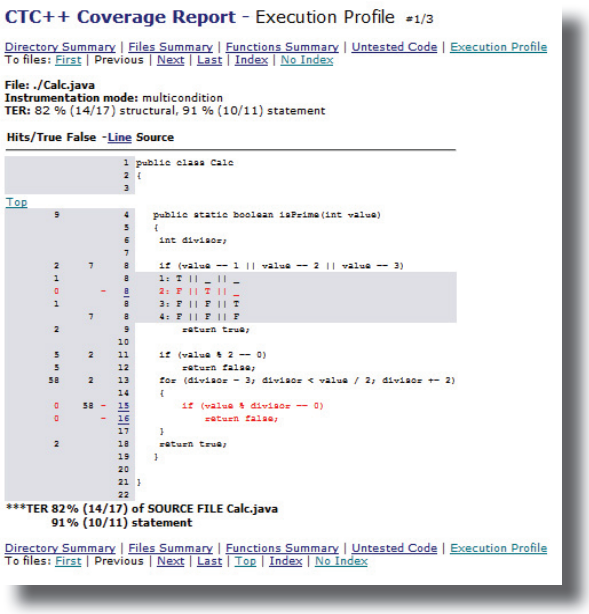
Easy to use

- Doesn‘t require debug build
- No modification to build files needed
- No modification necessary for existing sourcecode
- You will only need 4 little steps to test the coverage of your projects

How does it work?

1. Write your sourcecode
2. Call javactc instead of your default compiler
3. Run your application as normal
4. Use our tools like ctc2html, ctc2excel to create reports




CTC++ Coverage Report - Execution Profile #1/3

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile
To files: First | Previous | Next | Last | Index | No Index

File: ./Calc.java
Instrumentation mode: multicondition
TER: 82 % (14/17) structural, 91 % (10/11) statement

Hits/True False -Line Source

```
                1 public class Calc
                2 {
                3
Top
        9       4   public static boolean isPrime(int value)
                5   {
                6    int divisor;
                7
    2   7       8    if (value == 1 || value == 2 || value == 3)
    1           8    1: T || _ || _
    0       -   8    2: F || T || _
    1           8    3: F || F || T
            7   8    4: F || F || F
    2           9        return true;
               10
    5   2      11    if (value % 2 == 0)
    5          12        return false;
   58   2      13    for (divisor = 3; divisor < value / 2; divisor += 2)
               14    {
    0  58   -  15        if (value % divisor == 0)
    0      -   16            return false;
               17    }
    2          18    return true;
               19   }
               20
               21 }
               22
```

***TER 82% (14/17) of SOURCE FILE Calc.java
91% (10/11) statement

Directory Summary | Files Summary | Functions Summary | Untested Code | Execution Profile
To files: First | Previous | Next | Last | Top | Index | No Index

Get your free evaluation now.

Verifysoft Technology GmbH • In der Spöck 10-12 • 77656 Offenburg • Germany • Tel: (+49) 781 127 8118-0