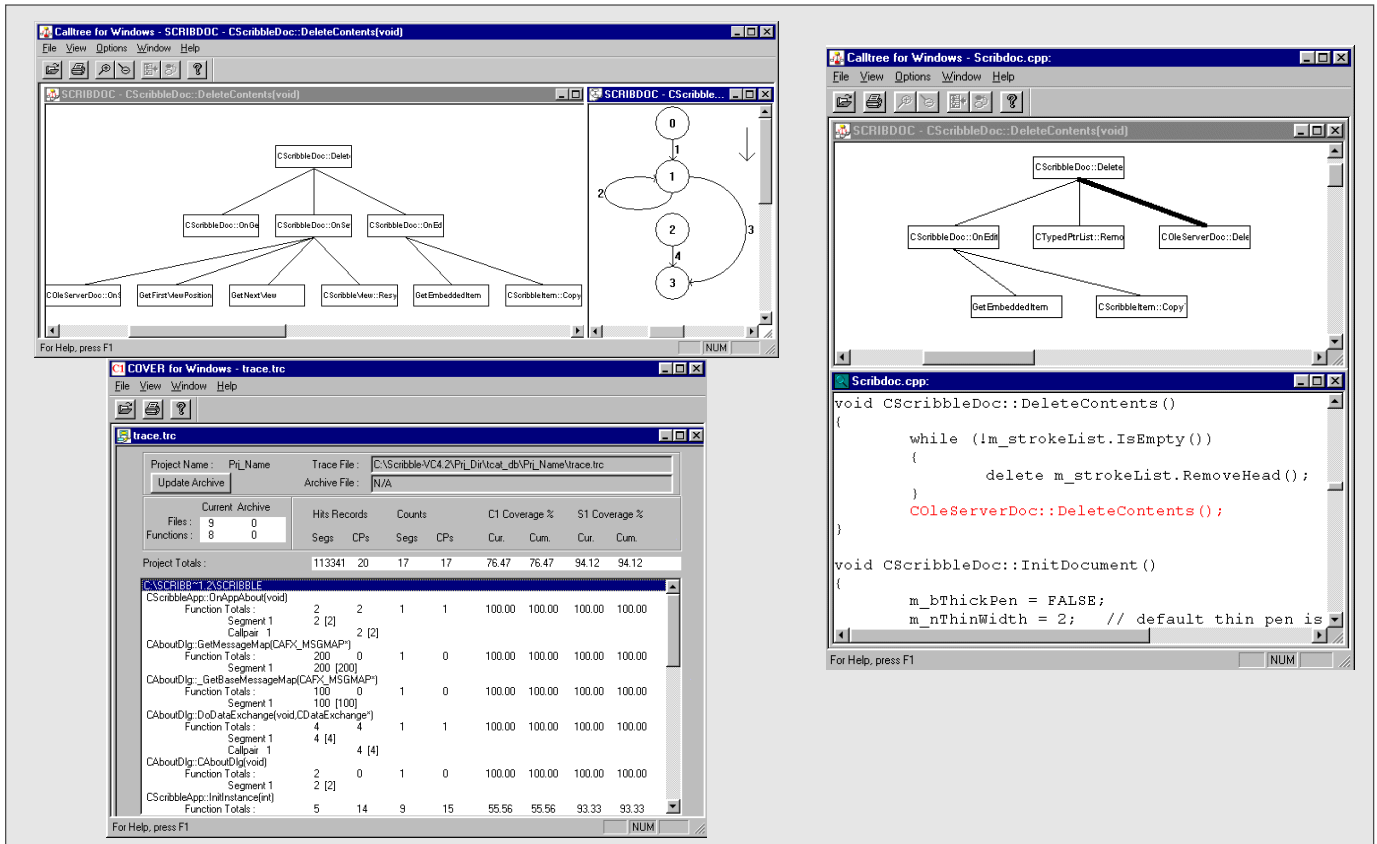




SOFTWARE RESEARCH, INC.

# TCAT C/C++™ for Windows™

## TEST COVERAGE ANALYSIS TOOL



Shown are a coverage report, a digraph, a calltree, and their associated source code, from a module that has been exercised. Using TCAT C/C++ for Windows, you can go directly from a digraph/calltree node or edge to source code with a single button click.

### PRODUCT DESCRIPTION

TCAT C/C++™ for Windows identifies which parts of a program's code have been tested at the individual function, as well as the system level. It can be used for unit testing (C1 Coverage) as well as system interface (S1) coverage testing to determine if all the interfaces have been exercised.

#### C1 COVERAGE

TCAT C/C++ for Windows measures the number of times logical branches have been exercised for both True and False conditions using its C1 metric, which is a more comprehensive yet efficient method than statement coverage. Statement coverage indicates which statements in a program

have been exercised. This is a common coverage strategy, but does not guarantee that the program has been thoroughly exercised as there can be multiple actions in a single line, of which not all are exercised even though the line is reported as tested.

TCAT C/C++ for Windows instruments the program under test, placing markers at each logical branch. When test cases have been run against the instrumented code, the C1 metric collects data on the markers and stores the data in a trace file. It then extracts the trace file information to create coverage reports. The reports indicate which code remains untested or frequently tested and which test cases duplicate coverage. In addition to coverage reports, TCAT C/C++ for Windows also creates an archive

file which stores all cumulative test information.

#### S1 COVERAGE

TCAT C/C++ for Windows also places markers at each function-call during instrumentation. When test cases have been run against the instrumented code, the S1 metric collects data on the markers and stores the data in a trace file. It then extracts the trace file information to create coverage reports that show which calls remain untested, unused, or frequently tested and which test cases duplicate coverage. Along with coverage reports, it also creates an archive file which stores all cumulative test information.



## DIRECTED GRAPHS & CALLTREES

During the instrumentation process, directed graphs that show a module's control-flow structure are created. These graphical representations are one of **TCAT C/C++ for Windows'** fundamental advantages: they are easy to interpret and give the user a quick overview of the results prior to looking at detailed reports.

The directed graphs allow the user to visually view a logical branch's level of coverage with its unique annotation feature and then displays and (with a single button click) highlights the source code for that logical branch. The directed graphs can be used to identify unexercised code and to study the architecture of a module to aid in re-designing new test cases for unexercised branches.

Instrumentation also generates call-trees that represent the caller-callee structure of a program. Call-trees aid the user in understanding code because they organize and identify a program's modules and connecting function-calls in a hierarchy.

The call-tree shows a function-call's level of coverage completeness with its annotation and color overlays. When a function-call obtains low coverage, the user can navigate to its source code. Along with coverage, call-trees display subtrees of caller-callee dependencies relative to a specific module and generate directed graphs for individual modules for an understanding of a module's control-flow structure.

By using **TCAT C/C++ for Windows** to monitor a system's level of testing, the user is ensured that the proper modules and interactions are actually tested.

## INSTRUMENTATION

- *Selective Instrumentation* — Allows for the instrumentation of isolated modules or exclusion of modules that have already reached a certain level of coverage.

- *Make File Support* — Easily integrates instrumentation steps into user's existing make file.

- *Control Structures* — Recognizes and processes all control structures.

## RUNTIME SUPPORT

- *Cross-Development Environments* — Performs coverage in cross, remote and embedded development environments.

- *Multiple Concurrent Processes* — Supports multi-process programs.

- *In-Memory Reduction* — Optionally accumulates trace records in memory instead of being written to a disk.

## COVERAGE REPORTS

- *Test Results* — Reports past, current and cumulative test results.

- *Report Types* — Provides hit, not hit, newly hit, newly missed, and linear and logarithmic histogram reports.

## ANALYSIS THROUGH DISPLAYS

- *Directed Graphs* (Digraphs) — Uses graphic and color overlays to diagram a module's logical branch relationship between decision points (nodes).

- *Call Trees* — Uses graphic and color overlays to diagram a program's module dependencies.

- *Subtrees* — Isolates a subtree of a call-tree relative to a specific module that the user wants to further investigate.

- *Source Viewing* — Allows navigation from a graph to a logical branch in the source code.

## DIGRAPH/CALL TREE CUSTOMIZATION

- *Zoom* — Sets the zoom scale of a digraph/call tree, so the user can focus in on areas of interest.

- *Nodes* — Sets node shape, size and length-to-width ratio.

- *Logical Branches* — Sets logical branch ellipse eccentricity.

## MULTIPLE INTERFACES

- Graphical User Interface (GUI)
- Command Lines
- Menus

## LANGUAGE SUPPORT

- C
- C++

## SUPPORTED PLATFORMS

- WIN 95
- WIN 98
- WIN NT/2000
- WIN XP

## TECHNICAL SUPPORT

- Telephone hot-line assistance for installation and technical questions is available.

- Maintenance contracts provide continuing product support and upgrades.

For more information about **TCAT C/C++ for Windows** or any Software Research products, call or write:



### SOFTWARE RESEARCH, INC.

1663 MISSION STREET, SUITE 400  
SAN FRANCISCO, CA 94103 USA

PHONE: (415) 861-2800  
TOLL FREE: (800) 942-SOFT  
FAX: (415) 861-9801  
E-MAIL: info@soft.com  
Web Site: http://www.soft.com



**TOOL TRADEMARKS:** CAPBAK/MSW, CAPBAK/UNIX, CAPBAK/Web, CAPBAK/X, CBDIFF, EXDIFF, SMARTS, SMARTS/MSW, S-TCAT, STW/Advisor, STW/Coverage, STW/Coverage for Windows, STW/Regression, STW/Regression for Windows, STW/Web, TCAT, TCAT C/C++ for Windows, TCAT-PATH, TCATfor JAVA, TCAT for JAVA/Windows, TDGEN, TestWorks, T-SCOPE, Xdemo, Xflight, and Xvirtual are trademarks, or registered trademarks of Software Research, Inc. and the SR logo are trademarks of Software Research, Inc. All other systems are either trademarks or registered trademarks of their respective companies. METRIC is a trademark of SET Laboratories, Inc. and Software Research, Inc. and STATIC is a trademark of Software Research, Inc. and Gimpel Software. Java is a trademark of SunSoft, Inc.

Software Research, Inc. reserves the right to make changes without notice, and within its own discretion, to any of the information contained herein.