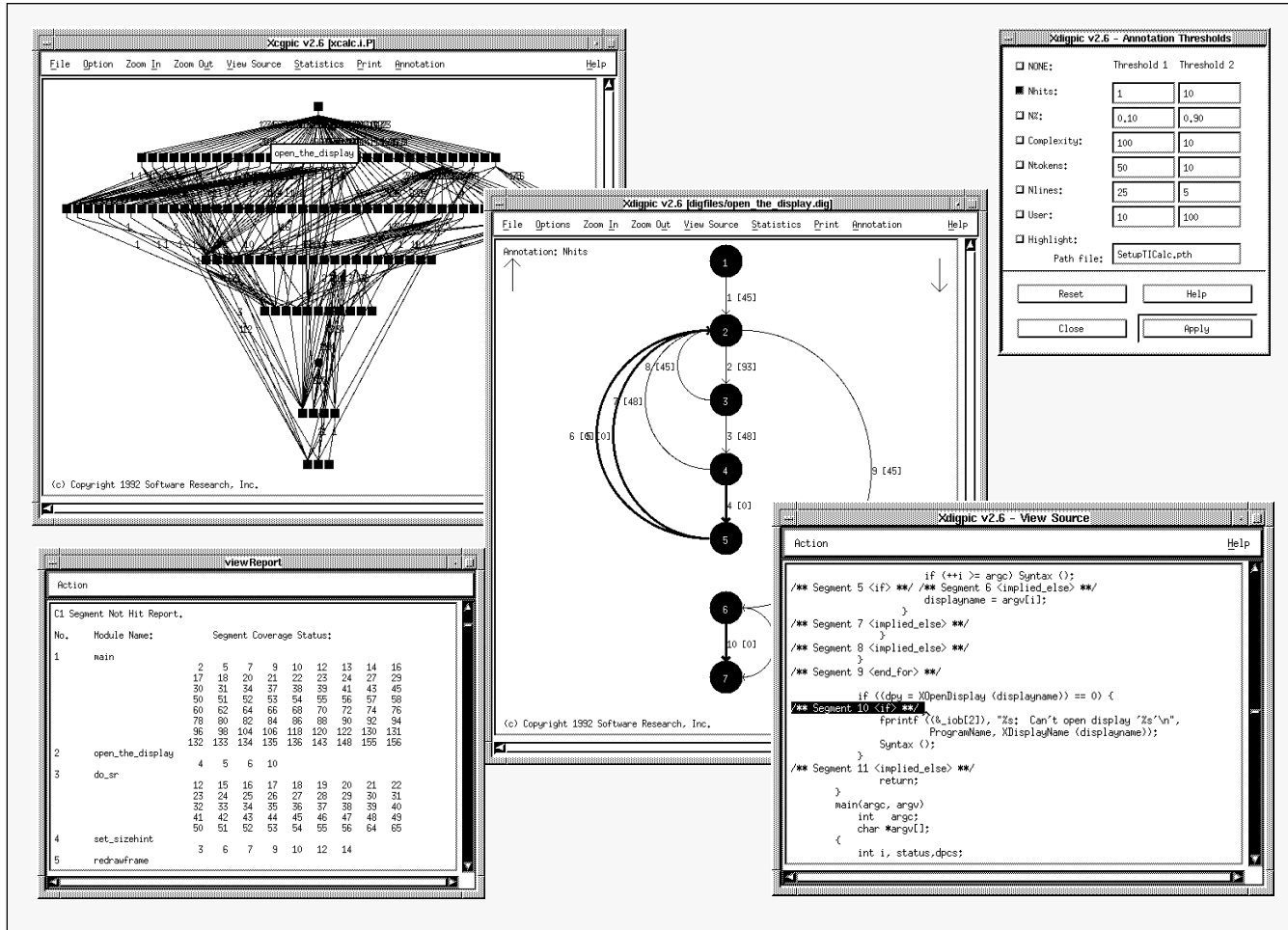




SOFTWARE RESEARCH, INC.

# TestWorks Coverage

TEST COVERAGE ANALYSIS TOOL SUITE



*STW/Coverage dynamically generates a program's call-tree (top left), a module's directed graph (center), and a static report of the unexercised logical branches (bottom left) with source code displayed for an un-hit logical branch.*

## PRODUCT DESCRIPTION

Coverage analysis acts as the vehicle to show where testing is incomplete and when it has been completed. While there is no way to determine if a test suite will reveal every defect in a program, it is possible to judge the quality of a test suite by measuring how test cases exercise code. A set of tests is only effective if it exercises a significant percentage of a system and, in particular, critical sections of the code.

**STW/Coverage**<sup>TM</sup>'s coverage analyzers give a numerical value to the completeness of a set of tests. They also show what parts of an application have been tested so that

effort can be focused on creating test cases that will exercise the parts that were not previously tested.

**STW/Coverage** measures runtime coverage at the following levels:

- **Logical Branch:** For unit testing; measures the number of times each branch has been exercised for both True and False conditions.
- **Call-Pair:** For integration and system tests; measures the number of times each function-call has been exercised, as errors in parameters are extremely common.
- **Path:** For critical functions; measures the number of times each path, which is a sequence of branches, was exercised.

Three analyzers and an observation tool are included in **STW/Coverage**: **TCAT**<sup>TM</sup> for logical branch analysis, **S-TCAT**<sup>TM</sup> for call-pair analysis, **TCAT-PATH**<sup>TM</sup> for path analysis, and **T-SCOPE**<sup>TM</sup> for dynamic visualization of coverage results.

**STW/Coverage**'s analyzers gather usage statistics on programs, as they are being exercised, and create coverage reports. Dynamically, **STW/Coverage** generates graphs which reveal the control-flow structure of a module and call-trees which show the caller-callee structure of a program. These displays show very quickly what is and what is not being exercised in a set of tests.



**COVERAGE METRICS**

- *C1 Metric* — TCAT provides logical branch coverage with the *C1* metric.
- *S1 Metric* — TCAT provides call-pair coverage with the *S1* metric.
- *Ct Metric* — TCAT-PATH provides path coverage with the *Ct* metric.
- *Complete Coverage* — STW/Coverage provides the only complete coverage solution available for test case validation.
- *Critical Coverage* — Provides coverage for mission-critical applications, such as an application needing FDA certification.

**INSTRUMENTATION**

- *Selective Instrumentation* — Allows for 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.
- *Path Viewing* — Displays the paths individually for a selected module.
- *Color Annotation* — Uses different color overlays based on lower and upper threshold values to indicate if a function-call or logical branch has been unexercised or has been heavily executed.
- *Source Viewing* — Allows navigation from a graph to a function-call, logical branch or logical path in the source code.
- *Statistics* — Summarizes information about the displayed call-tree or directed graph.
- *Print Option* — Prints directed graphs and call-trees to Postscript output.

**DISPLAY CUSTOMIZATION**

- *Zoom* — Sets the zoom scale of a call-tree or directed graph, so the user can focus in on areas of interest.
- *Functions* — Sets module size and length-to-width ratio in call-trees.
- *Nodes* — Sets node shape, size and length-to-width ratio.
- *Logical Branches* — Sets the ellipse eccentricity of logical branches.
- *Low-level Color* — Sets the color of the modules and their call-pairs or logical branches that are exercised less than a lower threshold value.
- *Mid-level Color* — Sets the color of the modules and their call-pairs or logical branch that are exercised between upper and lower threshold values.
- *High-level Color* — Sets the color of the modules and their call-pairs or logical branches that are exercised more than a upper threshold value.

**DYNAMIC VISUALIZATION**

- *Executed Program Parts* — Generates logical branch and call-pair coverage data in real-time.
- *Display Selection* — Shows coverage obtained for logical branches through directed graphs, call-pairs through call-tree displays, and an overall percentage through slider bars.

**MULTIPLE INTERFACES**

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

**MULTIPLE LANGUAGE SUPPORT**

- C
- C++
- Ada
- FORTRAN

**SUPPORTED PLATFORMS**

- Sun SPARC
- x86 Solaris
- SCO
- SGI
- IBM RS/6000
- HP 9000 - 700/800
- DEC Alpha
- UNIXWare

**TECHNICAL SUPPORT**



- Telephone hot-line assistance for installation and technical questions is available.
- Maintenance contracts provide continuing product support and upgrades.

For more details on **TestWorks Coverage**, contact:

**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.