Cleanscape Testwise
Software visulaization and analysis toolset

- Coverage analysis
  - *Static and dynamic software metrics*
  - *Produces high quality test sets*
  - *GUI coverage browser highlights executed code*
  - *Supports statement, decision coverage requirements*
  - *Supports bottom-up or top-down testing*

- Regression testing
  - *Identifies representative subsets of tests that should be re-executed*

- Advanced software maintenance
  - *Identifies code that is associated with features of the system*

- Dynamic debugging
  - *Automates redundant debugging tasks*

- Performance analysis
  - *Provides exact execution*
  - *Identifies only the relevant code*

- Dependency tracing
  - *Identifies date-sensitive objects*
  - *Language independent*

- File difference display
  - *Side-by-side color display*
  - *Thumbnail preview*
  - *Change statistics*

- Multi-Platform Support
  - *Unix, Linux, Windows*

- White Box Utility
  - *Allows Manipulation of hidden data and subprograms*
  - *Automatically generates test harnesses (Stub and Driver Programs)*

- Report Generator
  - *Provides Consistent Report Format Project-Wide*
  - *Execution History Shows Control Flow Between Units*

- Execution Manager
  - *Single or batch test case execution*
  - *Execution of modified test cases without re-compilation*
  - *Execute test harness under control of underlying compiler's debugger*

xSuds
Software Visualization and Analysis Tool Box

xAtac	
Test effectiveness measurement	
xRegress	**xProf**
Intelligent regression testing	Performance analysis
xSlice	**xFind**
Dynamic debugging	Static program dependency tracing
xVue	**xDiff**
Software maintenance	File difference display

Cleanscape Software Visualization and Analysis Toolset contains seven advanced software test automation tools that automate redundant testing tasks for more efficient software development and higher quality products

Adequate testing consumes about 70% of development resources in a typical software project. This is a key reason why quality often becomes the first casualty when software developers are faced with tight budget and approaching deadlines. Leading software developers like IBM and Telcordia Applied Research have learned how to achieve dramatic cost reductions and quality improvements by automatically analyzing the dynamic behavior of software with advanced software visualization and analysis tools. These software test automation tools are now available directly to software development organizations in Cleanscape Testwise.

Scrub your software clean

Testwise is a set of software testing, analysis, and understanding tools that streamline testing and maintenance of C, C++, and programs on UNIX, Linux, and Windows platforms. Testwise allows developers to integrate redundant test tasks into an automated software testing process that helps them identify and eliminate problems early in the development cycle. The Testwise tools can be used together or independently according to the needs of your specific project to help you automatically do the following:

- Determine code coverage and measure test effectiveness
- Conduct intelligent regression testing that minimizes regression test sets according to program coverage and execution costs
- Locate features and identify feature interactions in applications
- Statically debug source code
- Dynamically debug applications
- Conduct detailed performance analysis to locate performance bottlenecks
- Statically trace program dependencies
- Compare differences between files and applications

Software Visualization and Analysis Toolset

Cleanscape Testwise

Tools

*x*Atac Test effectiveness measurement tool

Key Functions

- Determine how much of your code is currently tested
- Create tests
- Determine what is missing
- Identify redundant test cases
- Determine whether product testers are finding bugs that have been found and fixed by developers
- Improve the software testing process

Description

*x*Atac is a test effectiveness measurement tool that greatly improves testing through targeted test creation and precise coverage measurement. This key utility helps programmers easily conduct thorough unit tests early in the software development process when problem eradication is less resource intensive.

*x*Atac Provides static and dynamic software metrics that software managers can use to control the development process.

*x*Atac Measures how well C, C++, and Ada programs have been exercised by a set of tests, identifying code within the program that is not well tested, and determining the overlap among individual test cases. Engineers and testers use xAtac to measure the adequacy of a test set and to identify areas of a program that require further testing. These measures indicate progress during the test phase of a project and can be used as acceptance criteria for subsequent stages of development and testing. System and regression testers use xAtac to identify a subset of a test set that achieves high coverage at limited cost.

*x*Atac provides visual feedback to users about the extent to which various parts of a program have been exercised and highlights areas whose coverage first will yield the most benefits. It not only helps testers to create high quality test sets but also guides them in creating small sets of high efficiency, high leverage test cases that yield high coverage quickly.

*x*Regress intelligent regression testing tool

Key functions

- Significantly reduce regression test costs
- Determine whether you are spending excessive resources in regression testing
- Understand how to select effective regression tests

Description

*x*Regress is an intelligent regression testing tool that allows developers to easily ensure that changes made to software, such as adding new features or modifying existing features, do not adversely affect features of the software that should not change.

Rather than rerunning all the tests in an expensive regression suite, *x*Regress gives developers an efficient means to identify a representative subset of tests that should be re-executed to revalidate modified software, significantly reducing testing efforts.

*x*Vue software maintenance tool

Key functions

- See where user functions are implemented in the software
- Know where features are implemented
- Visualize features in code
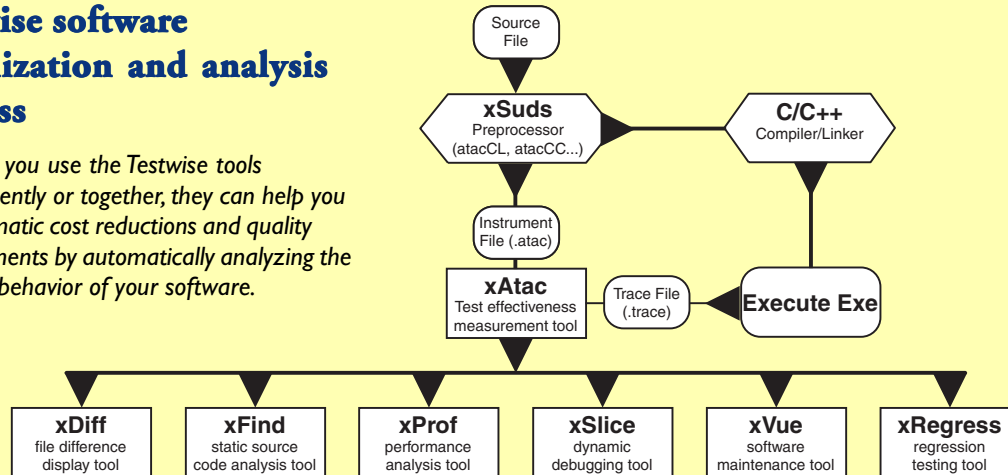- Reduce time involved with resolving reports

Description

*x*Vue is a software maintenance tool that allows a programmer to quickly locate code that is associated with features of the system. To determine where a feature is implemented in a program, one would run a small, carefully selected, set of tests — some which involve the feature and others that do not. Such tests are classified into three categories: invoking tests, excluding tests and don't_know tests. *x*Vue analyzes traces of program execution to look for program components that were executed in the invoking tests but not in the excluding tests. Other heuristics based on different combinations of invoking and excluding tests are also included to identify different sets of components.

*x*Vue provides visual feedback to users by highlighting feature-related source code. Depending on the tests and the heuristic selected and the structure of the program under examination, *x*Vue provide a good starting point for mapping features to program components that most existing software maintenance tools cannot. *x*Vue is highly effective at resolving trouble reports and in helping software maintenance engineers to quickly explore unfamiliar code.

Testwise software visualization and analysis process

Whether you use the Testwise tools independently or together, they can help you gain dramatic cost reductions and quality improvements by automatically analyzing the dynamic behavior of your software.

Source File

xSuds Preprocessor (atacCL, atacCC...)

C/C++ Compiler/Linker

Instrument File (.atac)

xAtac Test effectiveness measurement tool

Trace File (.trace)

Execute Exe

| **xDiff** file difference display tool | **xFind** static source code analysis tool | **xProf** performance analysis tool | **xSlice** dynamic debugging tool | **xVue** software maintenance tool | **xRegress** regression testing tool |

xSlice dynamic debugging tool

Key functions

- Pinpoint the location of faults from failures
- Locate bugs quickly
- Narrow down bugs to files, then functions, then lines of code

Description

*x*Slice is a dynamic program slicing debugger that automates tedious tasks that developers otherwise must perform manually while debugging their code. xSlice helps developers to focus on just the relevant code by eliminating the typical conceptual clutter of debugging. It makes the relevant pieces of the code stand out in no time with its intelligent analysis and the state-of-the-art graphical interface.

*x*Prof detailed performance analysis tool

Key functions

- Identify performance bottlenecks visually
- Improve the performance of your program
- Identify which part of the program slows execution
- Visualize the most frequently executed pieces in code
- Develop repeatable performance measurements

Description

Cleanscape *x*Prof is a program performance enhancement tool that helps developers to improve their code performance. While most profilers provide approximate clock times spent while executing code, *x*Prof provides exact execution found for various software items ranging from high level functions and subroutines down to the lowest level expressions. Such execution count based profiles provide software developers with a precise, repeatable, comprehensible way of measuring and improving code performance. *x*Prof uses an advanced graphical user interface to point out only the relevant code that programmers need to analyze, and possibly reorganize and/or rewrite, to improve the code's overall performance.

*x*Find static source code analysis tool

Key Functions

- Trace static program dependencies
- Identify date sensitivities in your applications
- Analyze difficult languages like C, C++, and Ada
- Identify redundant test cases

Description

*x*Find is a static program dependency tracing tool that helps software test engineers identify date-sensitive objects through a simple transitive relation.

Because the transitive relation is a simple heuristic, namely objects on the same line, *x*Find is language independent, and well adapted to pointer-based languages like C or C++. Other languages, like Perl or Tcl are also excellent candidates for analysis through *x*Find.

xDiff file difference display tool

Key Functions

- Display program differences
- Visualize the difference between two files
- Compare versions of code, documents, data, computer output

Description

*x*Diff is a tool for displaying differences between files. It displays two files side by side with line-by-line differences hilighted in color: a green background for lines that are changed; a red background for lines that are deleted, and; a blue background for lines that are added. Two customized bit-mapped scroll bars show a thumbnail sketch of the differences between the two files. These are useful for quickly locating changes, deletions, and additions. *x*Diff also reports the number of changes, additions, and deletions that have to be made to bring two files into agreement.

Cleanscape Testwise software visualization and analysis toolset

Target Applications

- Software programming
- Software test
- Software maintenance
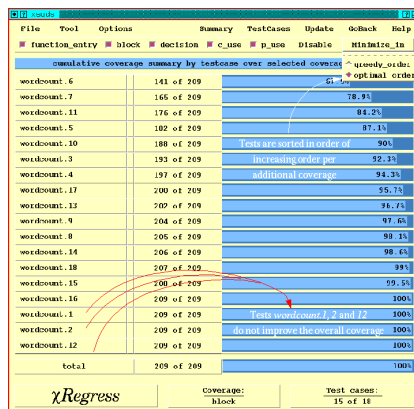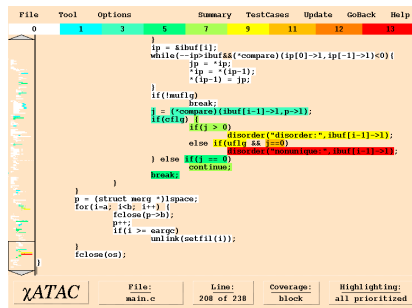
Specifications

Languages

- C
- C++
- Ada

Operating Systems

- SunOS
- Solaris
- AIX
- Windows
- UNIX
- Linux

Compilers

- Microsoft Visual C++ Versions 4.2, 5.0, 6.0
- IBM Visual Age for C++ for Windows Version 3.5 compilers
- RS6000 AIX 4.1.5 with C Set++ for AIX 3.1.3
- Sun4 SunOS 5.5 with C Set++ for Solaris Version 1.1.1
- Sun WorkShop Compilers C/C++ Version 4.2 for SunOS 5.5

Benefits

- Facilitates cross-platform by providing common test environment for Unix, Linux, and Windows
- Aid software development, selection and improvement of test sets
- Improve resource utilization
- Increases product quality and reliability
- Increases return on investment
- Maximizes productivity
- Shorten test and maintenance phases of software for faster time to market with higher quality product
- Easily standardize and automate established test processes
- Eliminate need to build test software (Stubs and Drivers)
- Makes software development faster, better, smarter, cheaper... *cleaner*

Software development just got a lot cleaner

THE name might seem new, but leading software developers in enterprise, government, and education have used our tools to automate software development for almost 20 years.

Today, Cleanscape's tools can automate your coding, analysis, debugging, documentation, testing, and build management for Linux, Unix, and Windows software development.

To see how clean your development can be, call or visit our web site for information about these tools for cleaner software development:

CLEANSCAPE
software international

Providing software professionals with tools and solutions that make the software development process faster, better smarter, cheaper...
Cleaner.

www.cleanscape.net

650 864-9600 **Main**
800 944-5468 **Sales**
sales@cleanscape.net

- **Cleanscape SourceMill.** Automatically generate commercial-grade code for virtually any language or platform from models.
- **Cleanscape Lint tools.** Analyze, document, and debug code and catch hundreds of problems your compiler will miss.
- **Test Automation Suite.** Generates tests, reports, and conduct coverage analysis for software components and systems.
- **qef.** Automate the software development process and replaces *make* with an advanced build management system.