

## Enabling High-Quality C/C++ Software, Automatically.

### What Is It?

Coverity Prevent SQS™ is the market-leading automated approach to identify and resolve the most critical defects in C, C++, and Java source code. By providing a complete understanding of your build environment, source code, and development process, Prevent SQS sets the standard in enabling high-quality software across organizations worldwide.

Prevent SQS for C/C++ automatically analyzes large, complex C and C++ code bases and detects critical, must-fix defects that could lead to system crashes, memory corruption, security vulnerabilities, unpredictable behavior, and performance degradation.

Prevent SQS features:

- **100% path coverage:** Prevent SQS for C/C++ analyzes 100% of the paths through your source code, ensuring that all possible execution branches are followed, while avoiding impossible paths to maintain fast execution.
- **Low false positive rate:** Prevent SQS for C/C++ maintains a very low false positive rate, ensuring that developers' time spent inspecting defects will result in noticeable quality improvements.
- **Highly scalable:** Prevent SQS for C/C++ analyzes millions of lines of code in a matter of hours, easily integrating into your regular build process with little or no additional hardware and no disruption to your development process.

### What Makes It Great?

Unlike other C/C++ analysis tools that focus on programming style and syntax-based checks, Prevent SQS for C/C++ performs *deep, interprocedural* analysis to uncover the *critical, must-fix* defects that matter most to developers. Prevent SQS for C/C++ leverages multiple analysis engines to uncover hard-to-find defects including:

- **Path Flow Engine** understands the control flow through each function in your code base, allowing Prevent SQS to analyze 100% of the paths through your code.
- **Statistical Engine** tracks behavioral patterns throughout your entire code base, allowing Prevent SQS to infer correct behavior based on previously observed behavior.
- **Interprocedural Summary Engine** enables Prevent SQS to perform a whole-program analysis of complex call chains at any depth across files and modules in a form that is most similar to the eventual executing Binary. This result in the highest-fidelity results available.
- **False Path Engine** solves each branch condition to determine if it will be true, false, or unknown on the current path. This allows Prevent SQS to efficiently remove obvious false positives from the set of defects reported.

### A sample of the critical defects reported by Prevent SQS for C/C++ include:

#### Concurrency Issues

- Double locks, missing locks
- Locks acquired in incorrect order
- Locks held by blocking functions

#### Memory Corruption and Mismanagement

- Resource leaks
- Calls to freeing functions using invalid arguments
- Excessive stack use in memory-constrained systems

#### Crash-causing pointer errors

- Dereference of null pointers
- Failure to check for null return values
- Misuse of data contained within wrapper data types

#### C++ Specific Errors

- Misuse of STL iterators
- Failure to de-allocate memory by destructors
- Incorrect override of virtual functions
- Uncaught exceptions

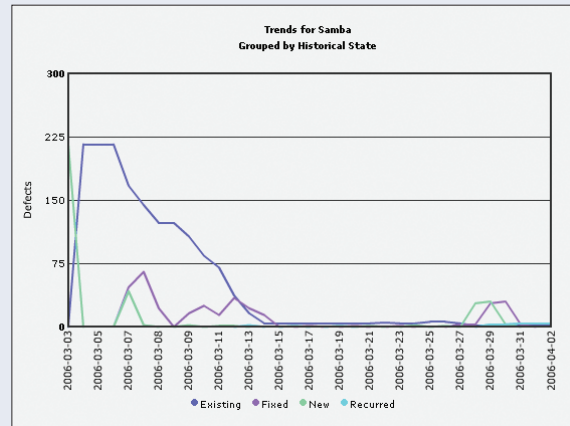
#### Window/COM Specific Errors

- Incorrect memory allocation with COM interfaces
- Incorrect type conversions

#### Security Vulnerabilities

- Buffer overruns
- SQL injection
- Cross-site scripting
- Integer overflow

Prevent SQS analysis results are committed to the central Defect Manager which provides automatic defect assignment and email notification whenever a new defect is detected.



Prevent SQS provides graphical reports with up-to-date information on current and historical defect status, resulting in instant visibility into code quality.

### Supported Platforms

- Windows
- Linux
- Solaris
- HPUX
- Mac OSX
- FreeBSD
- NetBSD

### System Requirements

- 1 GHz CPU
- 1 GB of RAM minimum, 2 GB recommended
- 1 GB of free hard disk space

### Supported Compilers

- gcc, g++
- Microsoft Visual C++
- Sun C/C++
- ARM
- Wind River
- TI Code Composer
- Green Hills
- Freescale/Nokia CodeWarrior
- IAR
- HI-TECH PICC
- Intel C/C++
- HPUX aCC
- Marvell MSA
- Support for other ANSI C/C++ compatible compilers available upon request

### Supported IDEs

- Eclipse 3.2 and later
- Eclipse-based IDEs including:
  - ARM
  - IBM Rational
  - Mentor Graphics
  - Nokia
  - QNX
  - Telelogic
  - Wind River

## Free Trial

Coverity offers a free trial of Prevent SQS that will detect a wide range of crash-causing defects in your code base within hours. No changes to your code are necessary, there are no limitations on code size, and you will receive a complimentary report detailing actionable analysis results.

Register for the on-site evaluation at: [www.coverity.com](http://www.coverity.com)

## About Coverity

Coverity ([www.coverity.com](http://www.coverity.com)) is the market leader in improving software quality and security. Coverity's groundbreaking technology automates the approach to identifying and resolving critical defects and security vulnerabilities in C/C++ and Java source code. More than 300 leading companies have chosen Coverity Prevent SQS because it scales to tens of millions of lines of code, has the lowest false positive rate in the industry and provides total path coverage. Companies like

Ericsson, HP, Samsung, EMC, and Symantec work with Coverity to eliminate security and quality defects from their mission-critical systems.

Coverity is a registered trademark; Coverity Extend, Coverity Prevent and Coverity Prevent SQS are trademarks of Coverity, Inc. All other company and product names are the property of their respective owners.



### Coverity Inc. Headquarters

185 Berry Street, Suite 2400  
 San Francisco, CA 94107 USA  
 US Sales: (800) 873-8193  
 International Sales +1 415 321 5237

### Coverity in Action:

[www.coverity.com](http://www.coverity.com)  
[scan.coverity.com](http://scan.coverity.com)  
[sales@coverity.com](mailto:sales@coverity.com)