CERT® C COMPLIANCE ENFORCEMENT

AUTOMATED SOURCE CODE ANALYSIS TO MAINTAIN COMPLIANCE

SIMPLIFY AND STREAMLINE CERT C COMPLIANCE

The CERT C compliance module reports on dataflow problems, software defects, language implementation errors, inconsistencies, dangerous usage and coding standard violations quickly and efficiently. The CERT C compliance module is an optional add-on for the QA·C static analysis solution, providing enforcement of the CERT C Secure Coding Standard ensuring vulnerabilities including security-related defects and violations are detected. The CERT C compliance module provides an extension to the analysis and reporting capabilities of QA·C to directly highlight violations of the CERT C Secure Coding Standard, and combines error detection and security best practice with full integration within the PRQA product suite.

The CERT C compliance module provides an out-of-the-box configuration for QA·C, which eliminates the need to manually configure the tool to enforce CERT C rules, and includes additional checks to supplement the already extensive suite of QA·C analysis checks. The existing QA·C report templates are also enhanced to allow generation of reports that specifically show the compliance of a code base to the CERT C secure coding standard, to inform internal stakeholders or to use for audit purposes.

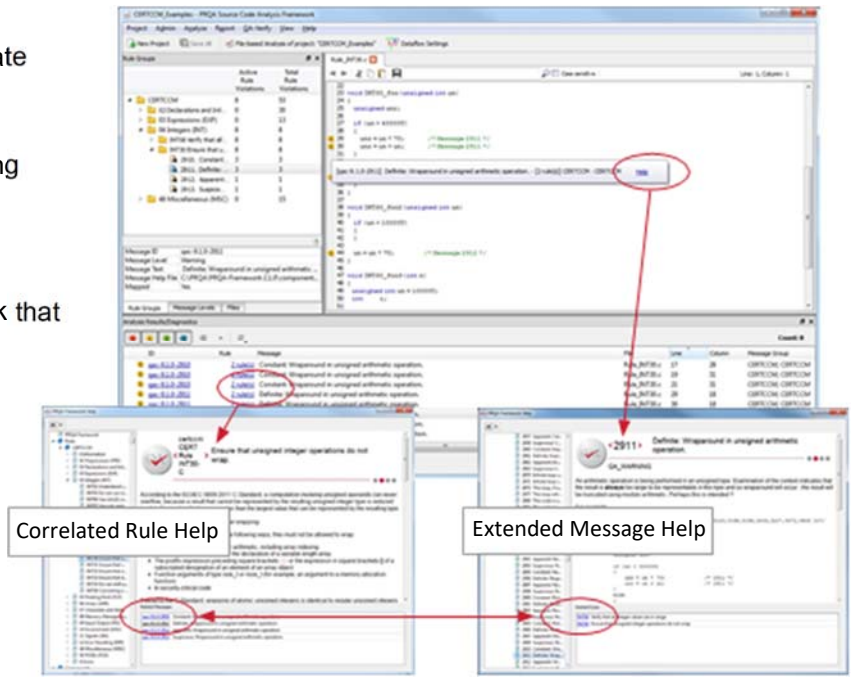
IDENTIFIES WHAT THE PROBLEM IS, EXPLAINS WHY IT'S A PROBLEM AND SHOWS HOW TO FIX IT

The QA·C static analyzer automatically performs in-depth analyses on your source code without executing programs. It checks your software for security vulnerabilities and conformance to CERT C secure coding best practices and can be configured to run locally on either desktop or server. QA·C identifies issues which compilers and most developers miss. These include lesser-known issues explicitly stated in the ISO standards and language constructs that, while not classified as incorrect, may result in unpredictable behavior.

Unlike bug catchers or less sophisticated static analyzers QA·C finds more issues while producing fewer false positives and negatives.

BENEFITS

- Automatically track, report and demonstrate CERT C Compliance
- Continuously inspect source code for conformance to the CERT C secure coding standard
- Scale to millions of lines of code
- Increase code portability and re-usability
- Give your developers contextual feedback that helps them correct and learn from mistakes
- Reduce bottlenecks caused by manual code review and slow analysis tools and methods
- Analyze your source code without executing programs



DON'T JUST FIND BUGS - ENABLE BEST PRACTICE

The goal of CERT C Secure Coding Initiative is to enable the development of safe, reliable, and secure systems, for example by eliminating undefined behaviors that can lead to undefined program behaviors and exploitable vulnerabilities and therefore result in high-quality systems that are reliable, robust, and resistant to attack. Developing secure code is different from developing function- ally secure code and may not be a well-understood concept by many developers. With this in mind, the CERT C Secure Coding Initiative attempts to educate developers and drive change rather than just document insecure code.

The CERT C secure coding standards consist of rules and recommendations, collectively referred to as guidelines. Rules are meant to provide normative requirements for code, whereas recommendations are meant to provide guidance that, when followed, should improve the safety, reliability, and security of software systems. Therefore, a violation of a recommendation does not necessarily indicate the presence of a defect in the code. Rules must meet specific criteria while recommendations are suggestions for improving code quality. The CERT C module helps your organization make informed decisions by finding and reporting on violations of both rules and recommendations covered within the CERT C secure coding standard.

KEY FEATURES

ADVANCED DEFECT PREVENTION

Using a proprietary, high-performance C language parser combined with a Deep Flow Dataflow analysis engine, QA·C is able to build an accurate model of the behavior of the software and track the value of variables in the code as they would be at run time. This sophisticated analysis approach maximizes code coverage while minimizing false positives and false negatives and allows QA·C to detect critical defects not reported by compilers or other tools and recognize issues caused by dangerous, overly complex and non-portable language usage.

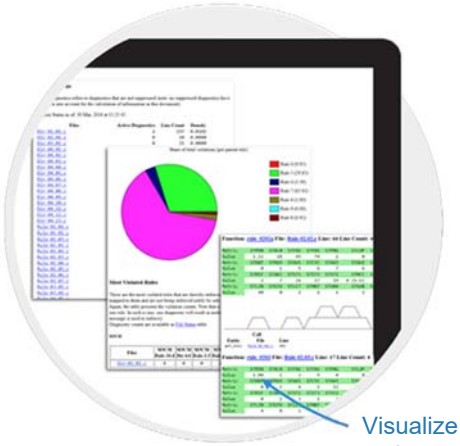
Identify unpredictable behaviours others miss

Clearly identify errors without executing code

ACTIONABLE RESULTS TO COMPLY TO THE CERT C SECURE CODING STANDARD

The CERT C module clearly identifies must-fix defects and includes a comprehensive knowledge base help system that provides detailed guidance with examples to support developers in fixing the issues found in the source code. Because developers get immediate and contextual feedback within their development environment, they can make the required changes as they are creating new code or reviewing existing code. In this way, developers build awareness of best practice approaches and can quickly form coding habits that are aligned with your organization's expectations.

MONITOR AND CONTINUALLY IMPROVE YOUR CODEBASE WITH CONFIGURABLE REPORTS



Visualize what parts of the code need the most attention

The compliance report helps you visualize which areas of your codebase require the most attention to reach a higher level compliance.

The code review report refocuses peer review on discussing design, optimization and meeting requirements rather than costly manual investigation of code conformance and correctness.

The suppression report provides information on message diagnostics that have been suppressed during analysis.

ANALYSIS OF INDUSTRIAL-SCALE CODE

Automated static analysis using QA·C assists in identifying defects, vulnerabilities, and compliance issues early in the development cycle where they can be fixed faster and at lower cost. QA·C is fast, non-disruptive, easy-to-use, and scales to any size of development environment. As a result, organizations whose products need to perform securely and reliably in mission critical and safety critical environments trust in QA·C to help lower the risk of software failures, improve quality and reduce time-to-market.

EASY TO LEARN AND EASY TO USE

The CERT C module functions as a plug-in within QA·C's powerful GUI and delivers a contextual drill-down environment linked to a deep knowledge base. QA·C explains why problems it discovers need to be corrected and then provides guidance to help in fixing them.

ADAPTABLE TO FIT EXISTING DEVELOPMENT ENVIRONMENTS

The CERT C module plugs into QA·C and is easily integrated into existing build systems and continuous integration environments to provide a means to enhance "early and often" testing with automated code analysis that helps to avoid errors that are expensive to fix late in the development cycle. This allows existing code review processes to be accelerated and refocused, thereby helping to increase overall productivity while also improving quality and security of the software. Additionally, the CERT C module and QA·C can be configured for incremental analysis to ensure that only new changes are analyzed and feedback can be provided quickly.

ROBUST AND FLEXIBLE CODING STANDARD ENFORCEMENT

The CERT C module is based on the CERT C Secure Coding Standard 2016 Edition, to automate compliance checks the rules specified in the standard with the addition of some rules from the Microsoft Windows, POSIX, API sections which are on the web- site but not included in the published standard. The module also automates the generation of the reports and audit documentation required to demonstrate compliance to the standard. QA·C functionality also allows messages to be suppressed at targeted source code locations and these suppressions can be included in deviation reports when required for audit to a specific standard.

KEY CHECKS

The CERT C compliance module helps to avoid constructs in the C language that can reduce code reusability and lead to product failures, functional safety issues and vulnerabilities that attackers can exploit. CERT C places particular focus on individual library functions that should not be called, correct usage of a library function, non-embedded and portable code, POSIX related best practices and threading related best practices.

The categories of CERT C rules and recommendations include:

- Preprocessor
- Declarations
- Expressions
- Integers
- Floating Point
- Arrays
- Characters/Strings
- Memory Management
- Microsoft Windows
- Input / Output
- Environment
- Signals
- APIs
- Concurrency
- Misc
- POSIX

TECHNICAL SPECIFICATIONS

GENERAL FEATURES

- Command line interface (CLI)
- Interactive GUI with message browser
- Online help & knowledge base
  - Usage & implementation contextual message
  - C language
  - CERT C coding standard
- Summary & detailed reports
- IDE integrations

CODE ANALYSIS FEATURES

- 1,700+ Selectable messages
- C language-specific parsing engine
- Parses code of any size & complexity
- Handles common language extensions
- Cross module analysis (link time checking)
- Semantic error detection
- Dataflow error detection
- Close name analysis

MESSAGE OUTPUT CONTROL

- Comment based suppression
- Baselining

RESULTS OUTPUT

- Configurable HTML reports
- Standard report types
  - Compliance
  - Code review
  - Suppression
  - Metric data

CODING STANDARD ENFORCEMENT

- Enforces 134 CERT C rules and recommendations
  - 50 rules
  - 84 Recommendations
- Rule subsets for legacy code
- Best practice issues
- Naming convention checker
- Layout checker
- Defensive programming - defect avoidance
- Extensible rule base
- Customizable message text
- Deviation support

QA Systems and Programming Research Ltd

QA Systems is an authorised reseller of the QA·C / QA·C++, QA·Verify static testing tools and their compliance module add-ons, which are owned by Programming Research Ltd.

QA·C, QA·C++ and QA·Verify, offer the closest possible examination of C and C++ code. All contain powerful, proprietary parsing engines combined with deep accurate dataflow which deliver high fidelity language analysis and comprehension. They identify problems caused by language usage that is dangerous, overly complex, non-portable or difficult to maintain. Plus, they provide a mechanism for coding standard enforcement.

Contact Us

For further information regarding QA·C, QA·C++ and QA·Verify and compliance module add-ons, please contact QA Systems at **info@qa-systems.de.**