Developing high-quality software is tough. ECLAIR is designed to help development, QA, and safety teams reach their quality goals.

MC3: MISRA C:2012 + Essentials

The *ECLAIR MC3* package is a combination of several of the many applications that run on top of ECLAIR, a powerful platform for the automatic analysis, verification, testing and transformation of C and C++ programs. This particular package combines:

- a state-of-the-art, medium-weight static analyzer that almost completely automates the assessment of compliance with **MISRA C:2012 Revision 1**, **BARR-C:2018**, **AUTOSAR-C:2009**, as well as other, complementary coding rules;
- a precise and flexible implementation of the source code metrics defined by HIS;
- the **ECLAIR Bug Finder**, a very fast static analyzer able to detect bugs and weaknesses that can lead to crashes, misbehaviors, and security vulnerabilities.

1 Highlights

- Proper coverage of MISRA C:2012, not just MISRA-C:2004 in disguise: many rules are radically different and require different checkers.
- No time wasted in writing compiler personality files (often of questionable correctness).
- Automatic production of accurate, faithful and (optionally) tamper-proof compliance reports.
- Easy-to-use yet powerful graphical user interface.
- Guideline violation and metric reports optionally available to the entire development team and management using web-based technology.
- Powerful mechanisms of differential reporting allow correlating changes in the code and the appearance/disappearance of violations (with possible interfaces to issue-tracking systems).
- No stress: free consultancy services for the initial configuration. This includes full assistance to help your company make the transition to the *MC3* package.

2 MISRA C:2012 Revision 1

MISRA C:2012 Revision 1 is the latest software development C subset developed by MISRA, which is now a de facto standard for safety-, life-, security-, and mission-critical embedded applications in many industries including aerospace, railway, medical, telecommunications and others.[1]

2.1 Coverage and Precision

The *ECLAIR MC3* package offers the most extensive, properly said MISRA C:2012 coverage available on the market, by providing support for around 98% of the guidelines.

Guidelines are enforced using very general and *accurate* checkers, which operate on the precise sequences of tokens and abstract syntax trees that are manipulated by the compiler. Coupled with the fact that ECLAIR always checks each guideline in the appropriate context (at the token, declaration, translation unit, whole program or whole system levels), this makes sure that the checkers for decidable rules are *exact* (neither false positives nor false negatives). For undecidable rules, ECLAIR's *MC3* package provides a medium-weight solution to the tradeoff among computational complexity, number of false positives and number of false negatives. In any case, when false negatives are possible, they are always clearly and unambiguously delimited.

Coverage of the MISRA C:2012 guidelines is summarized in the following table:

Support level	**#**
Fully supported (without false negatives)	157
Partially supported (with possible false negatives)	12
Under development	4
Total	173

3 BARR-C:2018, ECLAIR Bug Finder, and Other Essentials

The *Barr Group's Embedded C Coding Standard*,[2] BARR-C:2018, is, for coding standards used by the embedded system industry, second only in popularity to MISRA C. The adoption of the stylistic subset of BARR-C:2018 (79 out of 143 rules) can be part of complying with the MISRA requirement that a consistent programming style is adopted and systematically used as part of the software development process.[3] ECLAIR support for BARR-C:2018 has no equals on the market. The *ECLAIR Bug Finder* identifies security vulnerabilities, dead code, API misuses and other errors in C and C++ source code, including buffer overflows, dereferences of null pointers, pointer arithmetic errors, use of uninitialized variables, uninitialized or invalid return values, divisions by zero, undefined operations, dead stores, leaks of stack memory addresses, memory leaks, unreachable code, double-free, use-after-free, other dynamic memory allocation issues, lossy implicit conversions, excessive padding (memory waste), vararg functions mistakes, string manipulation errors, library API violations, insecure use of library functions, multithreading issues, dynamic type errors.

The *ECLAIR MC3* package includes dozens of other very useful services, among which are those supporting the AUTOSAR-C:2009 implementation rules for the development and maintenance of all AU-

[1] MISRA. *MISRA C:2012 — Guidelines for the use of the C language in critical systems*. Third edition, first revision. HORIBA MIRA Ltd, Nuneaton, Warwickshire CV10 0TU, UK, February 2019.

[2] M. Barr. *BARR-C:2018 — Embedded C Coding Standard*. Barr Group, 2018.

[3] R. Bagnara, M. Barr, and P. M. Hill. *BARR-C:2018 and MISRA C:2012: Synergy Between the Two Most Widely Used C Coding Standards*. embedded world Conference 2020 — Proceedings. WEKA FACHMEDIEN, Germany, 2020.

TOSAR *Basic Software* modules that are written in C, customizable naming rules, as well as additional software metrics.

4 Compliance Reports

ECLAIR can be configured to automatically produce compliance reports required to meet contractual obligations and industrial standards such as ISO 26262. The compliance report is obtained from the actual configuration, which, if properly done, will contain the reason for each deviation. Thus, carrying its rationale, any deviation goes straight from the configuration to the report.

In addition, thanks to ECLAIR's ability to intercept and fully understand the communication with the toolchain, the compliance report contains full details about the code and its analysis: which files have been compiled and/or analyzed (with full path and a cryptographic hash of their contents), the compiler/linker options, the full version of ECLAIR, ..., with even a cryptographic hash of the generated executables. All this allows the linking of the MCU's ROM actual content with the compliance report.

5 HIS Source Code Metrics

The HIS Software Test Working Group, recognizing the fact that software metrics provide an objective foundation to efficient project and quality management, specified a set of metrics to be used in the evaluation of software.[4] These metrics allow software quality to be assessed in terms of complexity, testability, readability, maintainability and so forth, and the quality of the software development process.

5.1 Coverage

ECLAIR's *MC3* package provides very precise and flexible coverage for all the HIS metrics with boundary limits: COMF, PATH, GOTO, v(G), CALLING, CALLS, PARAM, STMT, LEVEL, RETURN, VOCF, NOMV, NOMVP and ap_cg_cycle. If a limiting value for a metric is provided, ECLAIR can report where this value is attained and also, if needed, each subsequent point in the code where a value that breaches the limit is computed.

6 Proper Integration with the Toolchain

ECLAIR MC3, like all packages that run on ECLAIR, intercepts every invocation of the toolchain components (compilers, linker, assembler, archive manager) and it automatically extracts and interprets the options that the build system has passed to them. This allows for the seamless integration with any build system. Moreover, you do not need to engage in error-prone activities such as (a) specifying which files make up the application, and where the right header files are located; (b) configuring the static analyzer so that the analysis parameters match the options given to the compilers (several options *do* affect the program semantics); (c) writing down predefined macros and the architectural parameters such as sizes, alignment constraints, address spaces and so forth. All this is automatic and supports build processes that involve the automatic generation of source files that depend on the configuration, without the need to develop and maintain a separate analysis procedure: with ECLAIR the existing build procedure can be used verbatim.

[4] HIS source code metrics. Report HIS-SC-Metriken.1.3.1-e, Herstellerinitiative Software, April 2008. Version 1.3.1.

ECLAIR is available on most modern flavors of UNIX®, Linux, OS X® and Windows®, including Cygwin and MinGW, and can be used with just about any development environment. ECLAIR supports parallel and distributed program analysis, to leverage available computing resources. Most popular C/C++ compilers and cross compilers are supported, including ARM®, CodeWarrior™, Cosmic Software, CrossWorks™, GCC, Green Hills®, HighTec, IAR™, Intel®, Keil Software®, MPLAB®, Microsoft®, QNX™, Renesas Electronics, SOFTUNE™, TASKING®, Texas Instruments™, Wind River®, and clang/LLVM.

7 Graphical User Interface

All the verification tasks supported by ECLAIR can be specified and refined incrementally by means of a very convenient graphical user interface. This allows, for instance: finding coding rules using a powerful tag-based selection logic; activating and customizing coding rules, possibly restricting their use to only part of the project; selecting and customizing the kind of reports to be generated; defining project deviations and specific deviations (all deviations will in any case be reported into the final report); choosing to run the verification task immediately or save the task for later.

Detailed violation reports can be very conveniently browsed within the GUI. With a suitable license, violation reports can also be inspected using any web browser.

8 The Bigger Picture

ECLAIR is very flexible and highly configurable. It can support your software development workflow and environment, whatever they are.

ECLAIR is fit for use in mission- and safety-critical software projects: it has been designed from the outset so as to exclude configuration errors that would undermine the significance of the obtained results.

ECLAIR is developed in a rigorous way and carefully checked with extensive internal test suites (tens of thousands of test cases) and industry-standard validation suites.

ECLAIR is based on solid scientific research results and on the best practices of software development.

ECLAIR's unique features and BUGSENG's strong commitment to the customer, allow for a smooth transition to ECLAIR from any other tool.

ECLAIR qualification kits support tool qualification following the prescriptions of all major functional safety standards: CENELEC EN 50128 (railway), ECSS-Q-ST-80C (space), IEC 61508 (industrial), IEC 62304 (medical), ISO 26262 (automotive), RTCA DO-178C/DO-330 (aerospace).

Similar packages with MISRA-C:1998, MISRA-C:2004 and MISRA C++:2008 are also available!

For More Information

BUGSENG srl
Parco Area delle Scienze 53/A
I-43124 Parma, Italy
Via Lenin 132/F
I-56017 San Giuliano Terme (PI), Italy
Email: info@bugseng.com
Web: http://bugseng.com

bugseng

**no shortcuts,
no compromises,
no excuses:
software verification done right**