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

ECLAIR Bug Finder Package

The *Bug Finder* package is a general-purpose static analysis application for C and C++ that runs on top of ECLAIR, a powerful platform for the automatic analysis, verification, testing and transformation of C and C++ programs. The package includes a very fast static analyzer suitable for execution on the developer's desktop, which is able to detect and report bugs and weaknesses that can lead to crashes, misbehaviors, and security vulnerabilities.

1 Highlights

- Very fast: can be run on the developer's desktop, for early detection of a number of software defects.
- Extremely high signal-to-noise ratio.
- No need to write compiler personality files (time-consuming and often of questionable correctness).
- Automatically produces accurate and detailed reports.
- Immediate use from within most popular IDEs or *batch* use with reports stored in a database for later processing.
- Reports optionally available to the entire development team and management using web-based technology.
- Powerful differential reporting lets you correlate changes in the code with the appearance/disappearance of violations.
- No stress: free consultancy services for the initial configuration. This includes full assistance to help your company make the transition to the *Bug Finder* package.

2 Features

The ECLAIR *Bug Finder* is a tool designed to run on the developer's desktop as well as on integration servers. It enables the early detection of a number of software defects.

Avoid shipping defects to customers (with disastrous consequences and high remediation costs) and lower the cost of development by drastically reducing the resources spent in reworking and retesting.

Unlike other systems, ECLAIR *Bug Finder* does not require compiler- and target-dependent configurations (which are often error-prone) and presents accurate results that make it easy to identify "culprit" pieces of code and the correct remediations.

ECLAIR's *Bug Finder* package 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
- other common programming mistakes

Violation reports can be browsed using popular IDEs like Eclipse, Microsoft Visual Studio®, IAR Embedded Workbench®, Texas Instruments Code Composer Studio™, Keil µVision®, or any suitable editor. Violation reports can also be inspected using any web browser.

3 Proper Integration with the Toolchain

ECLAIR *ECLAIR Bug Finder*, 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.

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.

4 The Bigger Picture

ECLAIR is very flexible and highly configurable. It can support your software development workflow and environment, whatever they are. You can ask us to bend it to your precise needs or do that yourself.

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, such as Solid Sands SuperTest and the Plum Hall Validation Suite for C.

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

ECLAIR is developed by a passionate team of programming language and software verification experts.[1] Do not hesitate to let us have your feedback: you may be surprised to discover just how much your suggestions matter to us.

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

Several other ECLAIR packages can be integrated with the *Bug Finder*, such as those providing extensive coverage of all the MISRA C/C++ standards, including the newest security-related amendment to MISRA C:2012. For the development of security-critical applications, a *Common Weakness Enumeration* (CWE) package is 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**

[1] Among them is the Italian representative within ISO JTC1/SC22/WG14 (the international standardization working group for the programming language C), who is also a member of the MISRA C committee.