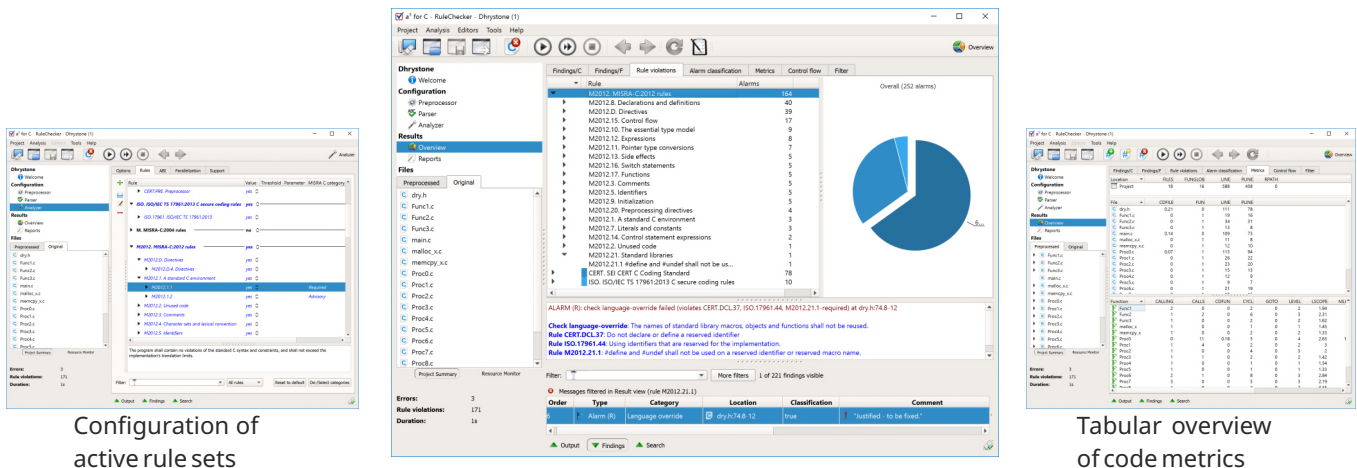


# RuleChecker

## Checking Coding Guidelines for C/C++ Programs

**RuleChecker** efficiently checks **coding guidelines** and computes **code metrics** for C/C++ programs. **RuleChecker** is an integral part of **Astrée**. A stand-alone version of **RuleChecker** is available under the name **QA-MISRA** from QA Systems.



### Why do you need RuleChecker?

**Safety-critical** or **mission-critical** software has to be developed according to coding guidelines to ensure **high-quality code**, and to reduce the risk of **programming errors** and **security vulnerabilities**.

- RuleChecker** is a static analyzer designed to check **coding guidelines** and compute **code metrics** for C/C++ programs. Multiple result views and graphical visualizations enable an **efficient result exploration**.
- Interactive **visualizations** of **static call graph** and **C++ class graph** help **program review and understanding**.
- RuleChecker** is **fast** and **easy to use**.
- RuleChecker** supports:
  - MISRA C:2004
  - MISRA C:2012 (incl. Ed. 3, AMD3, AMD4)
  - ISO/IEC TS 17961:2013
  - SEI CERT Secure C / C++
  - MITRE Common Weakness Enumeration (CWE)
  - MISRA C++:2008, C++:2023 Draft
  - Adaptive AUTOSAR C++14.
- RuleChecker** can be coupled with the sound static analyzer **Astrée** to guarantee **zero false negatives** and **minimal false positives** on semantical rules.
- RuleChecker** is fully batch-mode compatible and can be used in **continuous integration** frameworks. Open formats enable the analysis results to be automatically processed. A **Jenkins plugin** is available.
- The AbsInt Toolbox for TargetLink provides a seamless integration of **RuleChecker** with dSPACE **TargetLink**.
- Plugins for the **Keil µVision IDE** and **Eclipse** are available.
- RuleChecker** can be **automatically qualified** according to all relevant safety norms, including ISO 26262, DO-178B/C, IEC 61508, EN 50128, etc.

