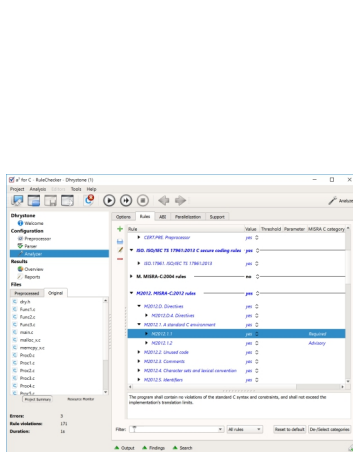


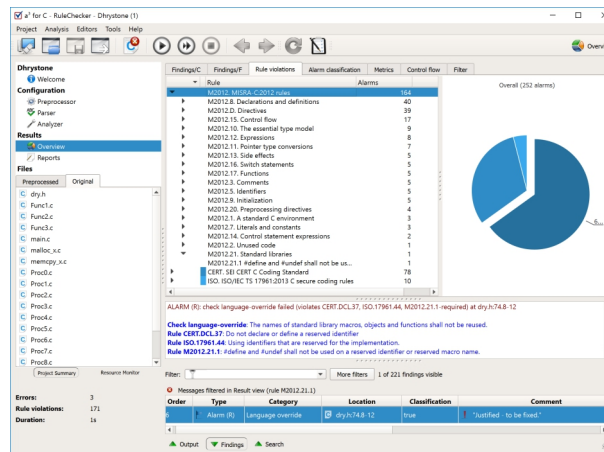
RuleChecker

Checking Coding Guidelines for C/C++ Programs

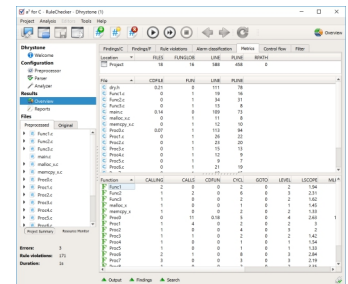
RuleChecker efficiently checks coding guidelines and computes code metrics for C/C++ programs.



Configuration of active rule sets



Result overview



Tabular overview of code metrics

Why do you need RuleChecker?

Safety-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 safety-critical C programs. Multiple result views and graphical visualizations enable an **efficient result exploration**.
- **RuleChecker** is **fast** and **easy to use**.
- **RuleChecker** supports:
 - MISRA C:2004
 - MISRA C:2012 incl. Amendment 1
 - MISRA C++:2008
 - ISO/IEC TS 17961:2013
 - SEI CERT Secure C
 - MITRE Common Weakness Enumeration (CWE).
- **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**. An **Eclipse** plugin is available.
- A flexible **client/server architecture** supports a wide range of rollout scenarios.
- **RuleChecker** can be **automatically qualified** according to all relevant safety norms, including ISO-26262, DO-178B/C, IEC-61508, EN-50128, etc.

For individual support of your in-house coding guidelines, please contact us.