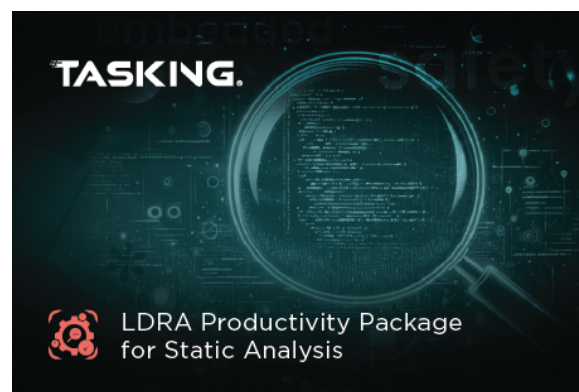


LDRA Productivity Package for Static Analysis

LDRA Productivity Package for Static Analysis

The LDRA Productivity Package for Static Analysis provides a specially targeted set of capabilities drawn from the LDRA tool suite to enable customers to comprehensively analyze software quality through static analysis. The productivity package enforces compliance with coding standards and clearly indicates software flaws that might otherwise pass through the standard build and test process. The automated static analysis and rules checking can be customized to support almost any in-house or universal standard.



Using LDRA's source code visualization, it's easy to identify exactly where code deviates from coding standards so it can be rapidly brought into compliance.

The package includes LDRA's best in class capabilities for improving the consistency and overall quality of code with a range of essential features designed to ensure compliance with safety and security coding standards. Static code analysis is used to evaluate code quality and gather critical quality metrics (or quality measures) such as cyclomatic complexity and HIS metrics. These can then be used to determine the level of clarity, maintainability, and testability of the code.

The screenshot displays the 'TBvision - LDRA Productivity Package for Static Analysis 10.2.1 © 2023 LDRA Ltd' application. The interface includes a 'File Explorer' on the left showing a project structure with files like 'safe_string.c', 'safe_sprintf.c', 'safe_compress.c', and 'main.c'. The main window is divided into two panes. The top pane, titled 'Code Review : Safe_Uilities : C - CERT-C Model', shows a table of violations with columns for 'Number Violated', 'Level of Violation', 'Phase Code', and 'Standard Code'. The bottom pane, titled 'Quality Review : Safe_Uilities', shows a table of quality metrics with columns for 'Value', 'Lower Limit', and 'Upper Limit'. The metrics are categorized by file and then by metric type (Clarity, Maintainability, Testability, Metric Groupings).

File	Metric	Value	Lower Limit	Upper Limit
Safe_Uilities	Clarity	84% Metrics Successful		
	Maintainability	100% Metrics Successful		
	Testability	98% Metrics Successful		
	Metric Groupings			
safe_string.c	Clarity	93% Metrics Successful		
	Maintainability	100% Metrics Successful		
	Testability	100% Metrics Successful		
	Metric Groupings			
safe_string_length	Clarity	70% Metrics Successful		
	Maintainability	100% Metrics Successful		
	Testability	100% Metrics Successful		
	Metric Groupings			
safe_string_compare	Clarity	70% Metrics Successful		
	Maintainability	100% Metrics Successful		
	Testability	100% Metrics Successful		
	Metric Groupings			
safe_sprintf.c	Clarity	86% Metrics Successful		
	Maintainability	100% Metrics Successful		

The bottom left pane shows a 'Log Window' with the following messages:

```

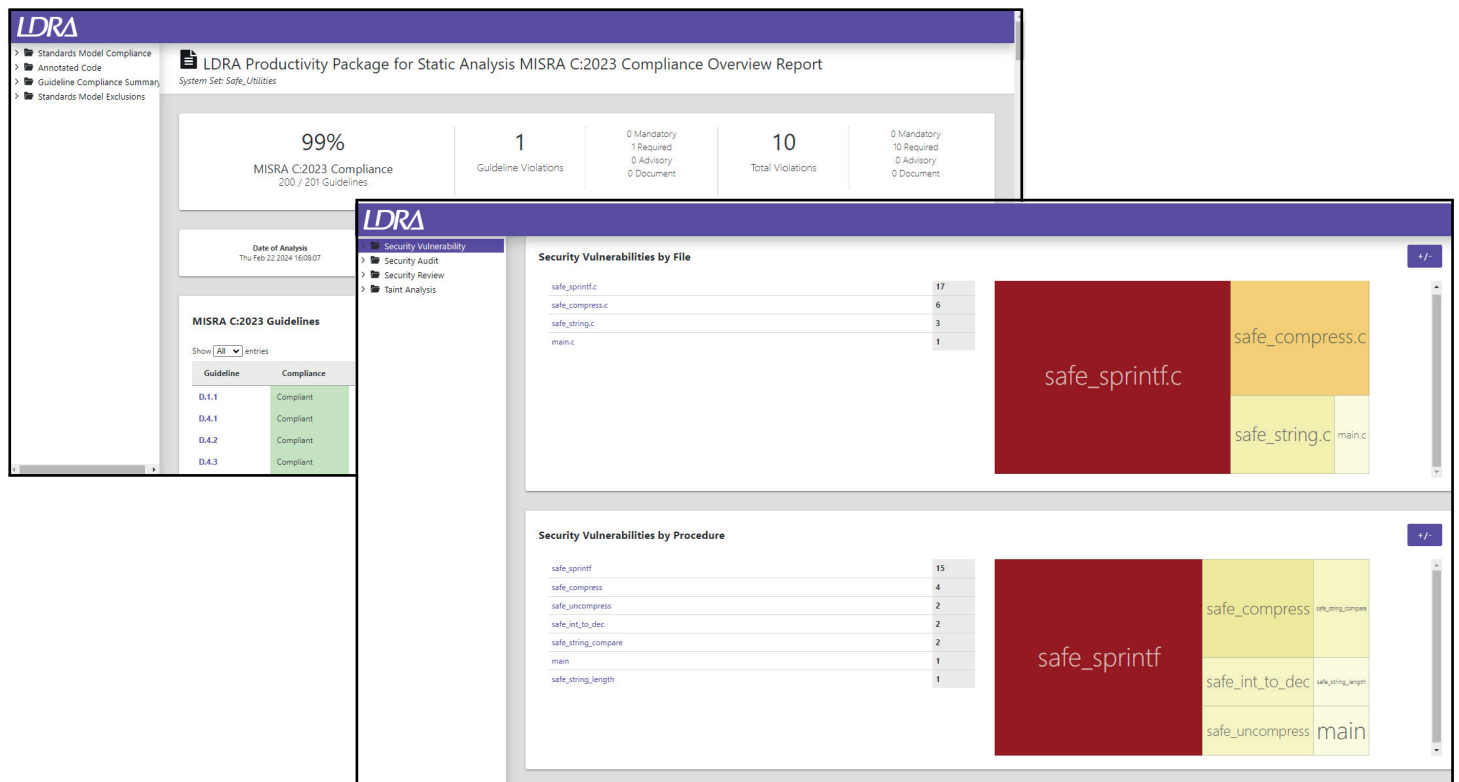
Command Line mode finished
Execution completed
Processing File G:\LDRA_Work\...
Processing File Completed
Code Review Started - Safe_Uti...
Code Review Completed
    
```

It also enables checking against the rules defined by safety and security coding standards such as MISRA, CERT, and BARR in addition to user-defined rules sets. CWE (Common Weakness Enumeration), while not strictly a coding standard, classifies and identifies security weaknesses in such a way that the static analysis tool can find them in the source code.

Exclusions, violations, and deviations from these standards may be justified and documented as permitted by quality guidelines such as MISRA Compliance document. This feature also allows graphical controls and regular expressions to be used by Developers to appropriately review standards and rule violations.

Reports are automatically generated from the analysis with appropriate HTML indexing for easy viewing and sharing across the development organization. These reports can be published to provide evidence of the fulfillment of certification objectives for standards including ISO 26262 and DO-178C. Optional add-on capabilities complement the base productivity package to provide a comprehensive toolkit for ensuring the highest level of assurance in the development of code, meeting the rigorous objectives of the chosen standard(s). Published reports can then be used as evidence to fulfill objectives within safety and security standards.

The flexibility and scalability of the package addresses the changing needs of the most demanding projects and enhances collaboration across large and small teams, making it ideal for use at any point in the software development lifecycle.



The LDRA Productivity Package for Static Analysis at a glance:

- Static code analysis including key quality, complexity, and testability metrics
- Coding standards compliance including MISRA, CERT, and CWE for both C and C++, along with many other commercially available coding standards
- The ability to define your own coding standards
- Flexibility and scalability to support projects at any point in the lifecycle
- Capabilities meet the needs of most software development and verification projects
- Reports for evidence-based objectives fulfillment of safety and security standards

TASKING®

EMPOWERING YOUR CODE, PULSE BY PULSE.

COMPILE | DEBUG | TEST

TASKING Germany GmbH



Streitfeldstrasse 19, 81673 Munich
Germany



+49 (0)89 2620 10080



www.tasking.com