LDRA tool suite® and Intland codeBeamer® ALM integration

LDRA, the leader in standards compliance, automated software verification, software code analysis, and test tools, has integrated its LDRA tool suite with Intland codeBeamer ALM to offer a complete Application Lifecycle Management and software testing solution for the embedded real-time, safety- and security-critical markets.

Intland codeBeamer ALM is designed to break down cross-enterprise silos and reduce lifecycle management complexity, helping development teams to build high quality products faster. The integration of codeBeamer ALM with the traceable, bidirectionally linked placeholders and artefacts of the LDRA tool suite extends its capabilities to embedded software development.

The unified solution enables traceability between requirements, source code, and the analyses and tests performed on host or target using the LDRA tool suite (Figure 1). The outcomes of these validation and verification activities provide insights both into the progress of development and into the code quality, complexity, and stability of the evolving system. These insights are reflected both in the Jira user interface, and in the evidential artefacts accessible from codeBeamer ALM or the LDRA tool suite.

Figure 1: This unified solution demonstrates traceability between requirements and development artefacts
Key Benefits

- provides an ALM framework for the software validation and verification of real-time embedded systems
- identifies code flaws using the LDRA tool suite, and logs, prioritizes, assigns and tracks them in codeBeamer
- delivers end-to-end bidirectional traceability between requirements, design, code and tests artefacts
- rapidly performs automated static analysis, dynamic analysis, unit testing, structural coverage, test case creation, and on-target hardware execution
- greatly increases the efficiency of software testing and reduces cost through automation and tool collaboration
- seamlessly performs post execution analysis and reviews with unique perspectives from both the LDRA tool suite and codeBeamer ALM
- supports qualification and certification in accordance with functional safety standards including DO-178B/C, ISO 26262, IEC 61508, and IEC 62304
- imports codeBeamer ALM requirements and test cases into the LDRA tool suite, complete with comprehensive traceability links
- exports LDRA tool suite test cases, verification status, and execution test results to codeBeamer ALM
- integrates MISRA, CERT and other coding standards into development environments that require evidential artefacts relating to safety or security standards
- automates test case, harness and stub generation for robustness testing
- exposes high-risk areas and defects by way of automated code analysis and report generation
- leverages codeBeamer ALM change and configuration management
- includes drag-and-drop functionality for traceability link creation
- provides Object Code Verification (OCV) for the most demanding safety-critical applications

LDRA TBmanager

TBmanager, the task-based interface component of the LDRA tool suite, provides a complete team-wide management workflow solution.

Leverage the integration by connecting to the codeBeamer web service, selecting any desired codeBeamer project, and exporting the multi-level requirements within the document files into TBmanager (Figure 2). Test cases and traceability links are retained, data is synchronized, and the most current records are always available to all stakeholders.

TBmanager provides the facilities to complete each test task, create test artefacts, and link them to the requirements in codeBeamer ALM. Data and results can be brought together to build various reporting views or perspectives which may include matrices and charts to show the progress of system, module/unit test, integration and regression testing. Code review reports, code coverage statistics, and code quality metrics like complexity, clarity and quality can also be included.
TBmanager is an efficient, easy-to-use solution facilitating complete system analysis. It collates the static analysis, dynamic analysis, unit, integration and system testing disciplines in one view, graphically illustrating end-to-end traceability and exposing any high-risk aspects of the software and its design.

TBmanager’s extensive and customizable objective mapping capabilities demonstrate adherence to functional safety and security standards, allowing project planning documents, assets and artefacts to be gathered and associated with objectives and requirements.

**Intland codeBeamer ALM**

codeBeamer ALM is a collaborative ALM platform designed to help meet functional safety goals and achieve compliance of medical device, avionics, automotive, and pharmaceutical embedded systems, yet accelerating the delivery of high-quality safety-critical products (Figure 3).
codeBeamer ALM brings transparency to collaborative, iterative Agile projects, drives delivery, lowers risks & costs, and reduces cycle times. Its capabilities cover the entire product development lifecycle, including the verification and validation processes associated with the LDRA tool suite (Figure 4).

codeBeamer ALM is scalable across teams, features, and infrastructure in an enterprise setting, integrating feedback and aligning all contributors across the DevOps lifecycle.

By monitoring performance and providing 360° traceability from idea to released product (Figure 5), codeBeamer ALM allows all contributors across the value chain to gain valuable lifecycle insights in support of decision-making. It lends clarity to iteration and release planning and management, and makes a “fail early, fix faster” approach easy to implement. Kanban scheduling provides for task management, and KPI charts including burndown aid Agile performance analysis.
codeBeamer ALM’s informative and graphical dashboard provides feedback on the project (Figure 6), and where life is made challenging by complex product lines with multiple product variants, codeBeamer ALM brings efficiency by easing reuse and avoiding redundant work.

Preconfigured templates and “baked-in” best practices help to hit the ground running, enhancing product quality, enforcing the use of mature processes, and simplifying safety-critical audit preparations. codeBeamer ALM provides a single source of truth for all contributors and auditors, reducing both the time and cost overhead associate with product verification, regulatory compliance, and tool validation.

**Code quality and verification workflow**

The LDRA tool suite’s static analysis of C, C++, Java or Ada source code can be used to check adherence to a selection of coding standards, including the MISRA language subsets and custom, user defined standards. The connected world makes LDRA’s support for security-focused coding standards from CWE, CERT and MISRA especially significant.

The dynamic analysis capabilities of the LDRA tool suite report on structural coverage, providing evidence that that code has been adequately tested during system and unit testing. Metrics including statement, branch, and MC/DC coverage, meet the demands of functional safety standards such as DO-178B/C, IEC 61508, ISO 26262 and IEC 62304. The provision of object code verification ensures that even the most exacting demands of compliance with aerospace standard DO-178C Level A are met.

The automation delivered by the LDRA tool suite’s unit and system-level testing dramatically reduces the time and cost of testing and eliminates the need for manual scripting. Unit, module, and integration test harnesses are seamlessly created by using the TBrun® component of the tool suite, leveraging the static analysis of procedure calls, unit interfaces, global and local parameters, return values, variable types, and data flow. Functions, methods, constructors, etc. are automatically stubbed, and the resulting harnesses are presented in TBrun’s user interface.

Tests vectors are then specified, and the tests executed on the host and/or target hardware (Figure 7).
TBextreme extends TBRun by providing a fully automated solution for test vector generation. Primarily used for robustness testing, its capabilities include the automatic generation of both “inside” (min, mid and max) and “outside” (invalid and out of range) boundary test case values to minimize the overhead associated with bottom up testing.

Conclusion

The integration of the LDRA tool suite and Intland codeBeamer ALM seamlessly automates complex testing tasks, accelerating time-to-market for embedded, critical software in the aerospace and defence, automotive, medical, industrial control, rail transportation, and energy industries. The integration helps ensure compliance and conformance with standards including DO-178B/C, DO-331, ISO 26262, IEC 61508, IEC 62304, EN 50128, SAE J3061, ARP 4754A, and Future Airborne Capability Environment (FACE™).

The complementary capabilities of codeBeamer ALM and LDRA’s solutions combine to create a powerful tool in the development and test verification of critical software systems. Whether adhering to the V-model process or deploying an Agile, Spiral, Waterfall or DevOps software development approach, the combination of LDRA’s requirements-based test suite and codeBeamer ALM’s application life cycle management tools have a major role to play in the improvement of software quality and in the delivery of timely, cost effective, safe and secure products.

More information can be found at [https://ldra.com/intland](https://ldra.com/intland).