



LDRA

Product Brochure

Delivering Software Quality and
Security through Test, Analysis
and Requirements Traceability

For companies producing the highest standards of software and seeking concrete, measurable software improvements throughout the development lifecycle, the **LDRA tool suite** provides a comprehensive range of both static and dynamic software analysis, in addition to unit testing and requirements traceability.

The tool suite, which has an unrivalled pedigree of over 35 years in the software quality market, is a completely integrated solution enabling customers to build quality into their software from requirements through to deployment. Integrating the LDRA tool suite into the software development process has proven critical to the success of many safety-, security-, business- and mission-critical software projects, especially those projects with strict safety-critical process guidelines, such as DO-178B and IEC 61508.

The technologies supported by the LDRA tool suite have been deployed across hundreds of projects in many vertical industries such as avionics, military, automotive, telecoms, energy and medical. These technologies can be utilised by an entire project team within the software development lifecycle, including project or requirements managers, developers, QA managers, and test and maintenance engineers.



LDRA Lifecycle Automated Activities

Which lifecycle tasks does the LDRA tool suite automate?

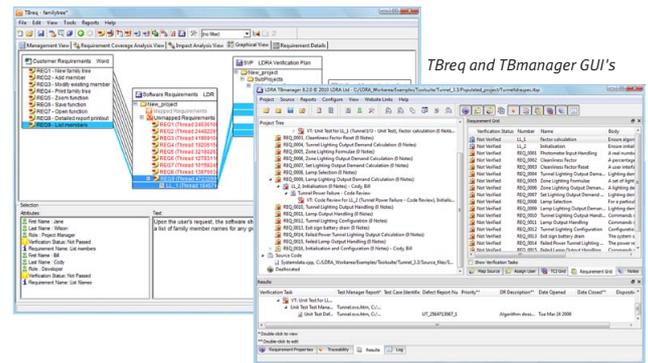
LDRA has developed its tool suite to assist with eight major tasks which need to be performed when producing software to the highest quality standards. In this capacity the *LDRA tool suite* is unique in its integration of requirements traceability with static and dynamic analysis, unit testing and its ability to support software testing on virtually any host or target platform.

LDRA has an open architecture, enabling easy integration with customers chosen tool chains for requirements management, modeling, software configuration management, IDE, RTOS, communication protocols, processors, compilers and target integrations.

1. Requirements Traceability Verification

The *LDRA tool suite* can be used to bridge the gap between requirements residing in any repository, such as IBM® Rational® DOORS®, Visure IRQA, IBM® Rational® RequisitePro®, and a Microsoft® Word document or Excel spreadsheet, to create a Requirements Traceability Matrix (RTM). This RTM may then be verified by the *LDRA tool suite* and mapped to LDRA analysed source code or flow down from mappings performed in modeling tools such as IBM® Rational® Rhapsody®, Artisan Studio® and The MathWorks Simulink®. With either scenario the requirements can be fully verified using an automated process that provides round-trip engineering across the entire software development lifecycle.

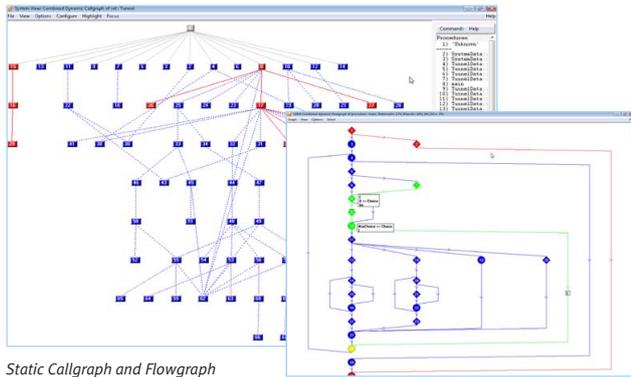
With the *LDRA tool suite* the reporting, resolution and management of defects are all complementary to requirements management, and these too can be incorporated into the RTM, bridging yet another major gap in most software development processes.



A key deliverable of the RTM, in conjunction with those of the LDRA tool suite, is to automatically ensure that if changes occur to the requirements, design or source code then these changes are easily traceable. The RTM capability, in conjunction with the other LDRA capabilities, allows users to greatly reduce the costs associated with verification, debugging and change management, which together contribute the largest aggregate cost in software development today.

2. Design Review

LDRA's design review performs an analysis of the structure, control flow and data flow of the source code under test, presenting the results graphically or textually and greatly enhancing design conformance (or non-conformance) review.



Static Callgraph and Flowgraph

Design visualisation at all levels improves understanding considerably and is an extremely powerful technique that helps developers to identify issues in code. The LDRA tool suite's colour-coded diagrams greatly enhance the design review process:

- Callgraph diagrams provide a hierarchical display of the application and system components.
- Flowgraph diagrams provide a graphical display of the control flow across program blocks.

The key deliverables of this component of the *LDRA tool suite* are a systematic analysis linked to a graphical representation of the as-built design, including system callgraphs and procedure flowgraphs, data flow analyses reports (from both parameter and variable usage viewpoints) and path level analysis reports employing LDRA's LCSAJ technology. These deliverables document a top-down design review process that can be used by development teams and management. The results from these design analysis are also automatically carried forward into LDRA's unit testing and test verification components.

3. Code Review

Powerful static analysis features in the LDRA tool suite, including programming standards checking and sophisticated reporting capabilities, help to minimise the need for repetitive and labour intensive manual code inspections.

To prevent runtime errors and ensure that source code satisfies the development team's coding standards, the user can run checks against sets of industry standard programming rules, or configure their own user-defined model. The industry recognised standards include MISRA-C:1998, MISRA-C:2004, HIS, JSF++ AV, High Integrity C++, MISRA C++:2008, CERT C or CWE. The code review facilities of the LDRA tool suite provide the user with the ability to quickly identify violations early in the software development lifecycle, often revealing latent errors that would not be identified through the standard testing process. This improves the quality of code, reduces the errors found during formal testing and helps to ensure on-time delivery within budget.

This automated, efficient and repeatable process saves both time and resources, and supplies developers with unparalleled levels of detail that cannot be matched by traditional manual techniques. The Code Review Report provides an overview of code violations and warnings with the ability to drill down into the source code to address the problems which have been identified.

Number Violated	Level of Violation	Standard Code
2	Required	MISRA-C:2004 8.7
2	Required	MISRA-C:2004 2.2
5	Required	MISRA-C:2004 8.5
5	Required (C)	MISRA-C:2004 13.13
9	Required (C)	MISRA-C:2004 8.5
	Advisory	MISRA-C:2004 6.3
	Advisory	MISRA-C:2004 5.7

TBvision GUI

4. Quality Review

This component enables the user to quickly determine the quality of the software using comprehensive sets of metrics encompassing software testability, maintainability and clarity. LDRA has applied significant effort to the automation of this highly beneficial quality review process.

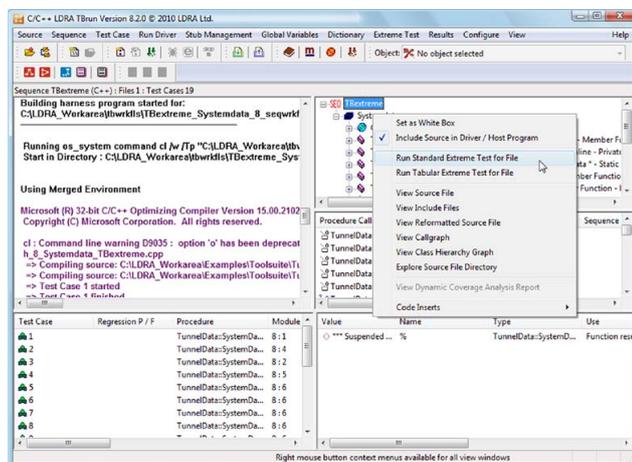
The LDRA tool suite goes beyond common complexity metrics to determine “essential complexity”, that is the sections of the source code which require refactoring. Code density metrics are another unique feature of LDRA's quality review process and these point to potential future maintainability problems.

The key deliverables of the quality review process are the quality model visualisation, system quality metrics and code refactoring guidelines that can all be used to improve overall code confidence. The consistent enforcement of a user specified quality model is reported for development team and management review thereby supporting and improving project level communication.

5. Unit Test

LDRA's unit testing tool assists developers to generate the test vectors, test harnesses and code stubs necessary to ensure complete testing for both unit testing and regression testing. Integrated coverage analysis provides the feedback necessary to ensure that enough test cases are defined to exercise all of the code under test before being handed over to system integration and test. This helps to root out defects in the code that may not be identified until later on in the product development lifecycle. Once the test cases are defined, the tool automatically generates a test harness containing the sequence of test cases that can be used on both the code under development and for regression testing. This code can then be run on either the host or target based system.

Compressed development schedules often prohibit the manual generation of a complete set of unit test cases. LDRA's extreme testing capability helps to ensure that the essential job of unit testing is completed by using the knowledge of the code structure gained from the static analysis to automatically generate test vectors with the objective of exercising as much of the code under test as possible. Once generated, test harnesses for test case execution and regression testing are then produced, eliminating the traditional time and resource problems associated with bottom-up testing.



Unit Test GUI

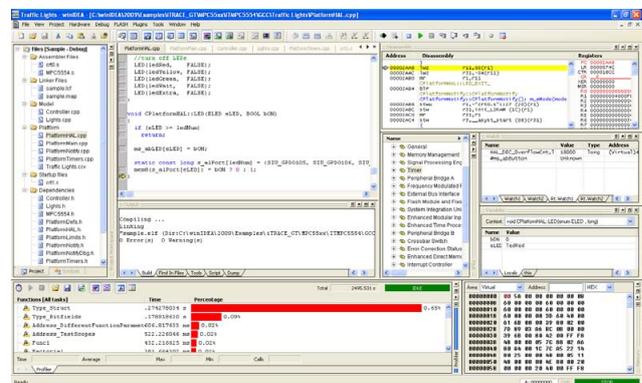
Additional key benefits of LDRA's unit test capability include:

- Automated test vector and driver / harness generation with no manual scripting requirement
- High levels of test throughput may be achieved via the intuitive graphical and command line interface options
- Sophisticated automated analysis facilities reduce test effort, freeing up developers and empowering testers
- Test data and test results are stored and maintained for fully automated regression analysis
- Automated detection and documentation of source code changes
- Tool driven test vector generation
- Facilitates execution of tests in host, target and simulator environments
- Automatically generated test case documentation including pass/fail and regression analysis reports

6. Target Testing

A key feature of the target testing capabilities is the high degree of flexibility that the tool supports, which virtually eliminates risks associated with unique target constraints and evolving target systems. Moreover, the LDRA tool suite's Object Code Verification offers a proven, single-tool solution to directly compare code coverage at the source code level with that achieved at the object code level.

A key deliverable of the LDRA tool suite in respect of target testing is the ability to provide highly automated solutions for a wide range of processor / IDE (Integrated Development Environment) combinations, which greatly assists in overcoming the significant challenge of deploying the test execution on a specific target.



Example Target Test System

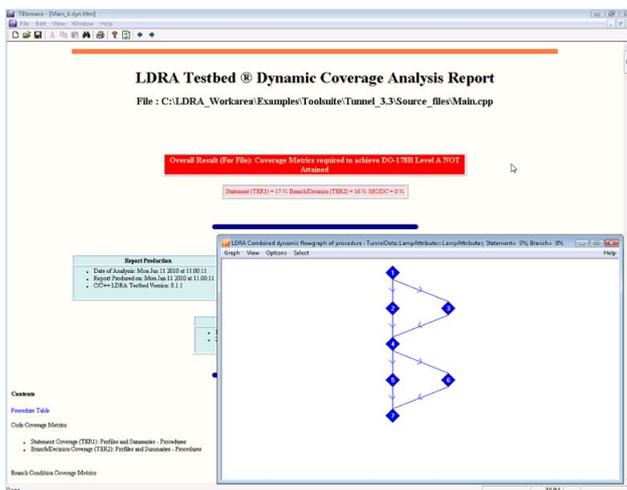
7. Test Verification

When building software to meet the highest quality standards, it is essential to ensure that the system is tested to completion. Coverage analysis measures test data effectiveness by measuring how much of the software code structure under test has actually been exercised. Structural coverage has become the staple technique for quality-conscious groups, such as avionics manufacturers, who must ensure that their products are ultra-reliable.

LDRA provides an extensive range of coverage analysis measurements in its LDRA Testbed and TBrn products, ranging from straightforward statement coverage to the Modified Condition / Decision Coverage (MC/DC) required for most safety critical avionic systems. Also included is Linear Code Sequence and Jump (LCSAJ) coverage that offers a more thorough test data assessment than decision coverage yet avoids the exponential difficulty of path coverage.

The DO-178B software development guidelines for avionics demand that for the most safety-critical applications (Level-A classification) coverage analysis of 100% to the MC/DC level be performed at the source code level, in addition to ensuring that 100% of the object code (assembler) is executed. The LDRA Object Box solution augments the source level coverage solution by adding object code coverage to provide a complete structural coverage analysis solution. Object Box comes in a number of variants to support different processor instruction sets. A typical example might see C/C++ and TMS320C2xx Assembler copies of the LDRA tool suite teamed together.

The 'Object-box Mode' enables users to create test cases for structural coverage of high-level source and apply these exact same test cases to the corresponding object code.



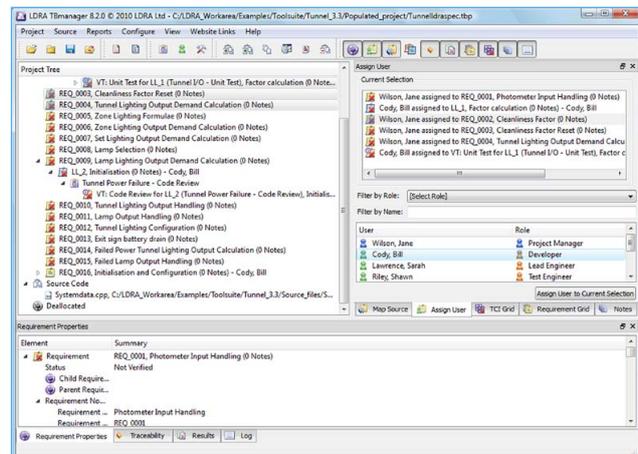
LDRA Dynamic Coverage Analysis Report

The ability to quickly pinpoint inadequately or untested software, test to a high standard and reduce regression testing costs is vitally important, helping to save resources, increase product quality and shorten time-to-market. The key deliverables of LDRA's test verification component are providing Coverage Metrics for the following:

- Statement
- Branch/Decision
- Procedure/Function Call
- Branch Decision Condition
- Branch Condition Combination
- Modified Condition / Decision Coverage (MC/DC for DO-178B Level A)
- Dynamic Data Flow
- LCSA)

8. Test Manager

Being able to fully understand a system can take a long time and require large amounts of resources. The LDRA tool suite assists developers and test engineers in understanding, documenting and maintaining large complex systems. Additionally, Test Manager automatically detects changes to the source file(s) and performs a validation of associated regression test cases.



TBmanager GUI

Some of the key features include:

- Automatic Software Change Detection which identifies and tracks significant source code changes and enables project teams to accurately monitor the impact of these changes on the testing process and the overall analysis of the code.
- Regression Test Management this provides the ability to have traceability information and accept the tests for automated regression testing. The tests and their results can be easily stored with their associated unit under a test baseline control system.
- Documentation which assists the process of achieving quality standards such as ISO 9001:2000. The level of detail within the reports includes a high level overview, quality report and metrics reports; and from within these reports, it is possible to drill down directly to the source code.

A key deliverable of the Test Manager is to provide full system documentation for audit purposes and version-control purposes, as well as helping to reduce ongoing maintenance costs.

LDRA Testbed®

At the heart of the *LDRA tool suite* is *LDRA Testbed*, providing the core static and dynamic analysis engines for both host and embedded software. *LDRA Testbed* enforces compliance with coding standards and provides clear visibility of software flaws that might typically pass through the standard build and test process to become latent problems. In addition, extensive test effectiveness feedback is provided through structural coverage analysis reporting facilities which support the requirements of the DO-178B standard up to and including Level-A.

TBvision®

TBvision presents code standard violations and software flaws in the context of the original source code. The interactive environment allows switching between reported violations; the original source code and any of the *LDRA Testbed* supported coding standards. In so doing *TBvision* clarifies to the user why an issue is being reported and what remedial action is required. In addition, software integrity can be measured and reported in terms of quality, security, or simply the presence of defects (including dynamic memory errors). *TBvision* presents the software flaws identified by static analysis from any of these perspectives and identifies the issues that need to be addressed to ensure that a software project meets its objectives.

Additional Options

TBmisra®

TBmisra provides coding standards checking against MISRA-C:1998, MISRA-C:2004, MISRA C++:2008, JSF++ AV & VSOS and High Integrity C++.

TBsecure®

The *TBvision* plug-in *TBsecure* provides secure code standards checking against the Carnegie Mellon Software Engineering Institute (SEI) CERT C secure coding standard and the Common Weakness Enumeration (CWE) dictionary.

TBsafe®

TBsafe has a set of analysis tools to help rigorously test code to exacting standards such as those required for DO-178B, Def Stan 00-55 and IEC 61508. Features include Information Flow Analysis, Dynamic Data Flow Coverage, Modified Condition/Decision Coverage (MC/DC) for DO-178B Level A as well as Exact Semantic Analysis.

TBeXtreme®

TBeXtreme revolutionises unit testing by using the information gathered by *LDRA Testbed* to provide a totally automated solution. *TBeXtreme* eliminates the traditional time and resource problems associated with bottom-up testing.

TBobject Box®

TBobject Box provides an Object Code Verification capability, offering the only direct way to relate code coverage at the source code level with that achieved at the object code level. In addition to this the tool provides the mechanism to extend, where necessary, the code coverage at the assembler level.

TBrun®

TBrun is LDRA's automated unit testing tool, providing a GUI driven interface for creating unit and module test cases for either host or target-based software. Test harnesses are automatically generated by *TBrun*, which also supports the creation of stubs for code that is outside the scope of the tests. Structural coverage analysis can also be included in the test harness, helping to ensure that the defined test cases execute 100% of the code under test. Formal test reports can be produced and, once generated, the test harnesses then serve as a regression test environment for future development phases.

TBreq®

For requirements driven projects *TBreq* supports the tracing of requirements through the entire software development lifecycle. High level requirements captured in standard office documents or third party RTM solutions, such as DOORS, IRQA or Requisite Pro can be traced through lower level documents to the code itself, and also to the associated verification activities and artifacts produced by the *LDRA tool suite*. The end result is a Requirements Traceability Matrix document that captures how each high level requirement was traced to the final software product, including the associated verification activities and results.

TBevolve®

The *TBvision* plug-in *TBevolve* enables project teams to accurately monitor the impact of code changes on their testing process. As the source code changes *TBevolve* will compare a baseline copy of a system with new versions and will highlight changed source code lines and report on untested source code which affects the overall code coverage analysis.

TBpublish®

TBpublish captures the analysis and test results from the *LDRA tool suite* and publishes the results via an HTML index, into a self-contained directory for easy navigation and collaborative reference.

TBaudit®

TBaudit offers development and test managers an automatically generated, user-configurable Microsoft Word report which contains the results of the Review and Testing activities carried out by the development, QA and testing teams using the *LDRA tool suite*. *TBpublish* is a prerequisite for *TBaudit*.

Target Licence Package (TLP)

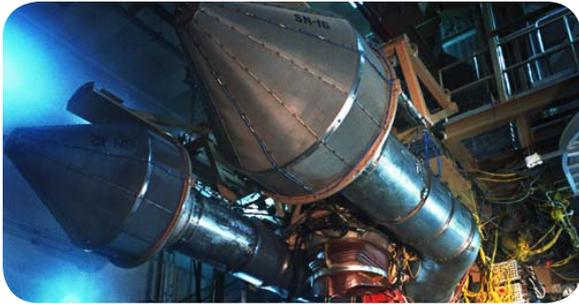
The *Target Licence Package* provides the right to use and to receive support for the *LDRA tool suite* when it is used for target testing. The TLP allows the user to receive assistance from LDRA in configuring the *LDRA tool suite* to interface with a specific embedded tool chain.

DO-178B Tool Qualification Support Packs

The *LDRA tool suite* falls under the category of 'Software Verification Tool' for DO-178B tool qualification purposes and the *LDRA Tool Qualification Support Pack* enables clients to follow a process which will provide the information required in order to qualify the *LDRA tool suite*.

Client Testimonials

Pratt & Whitney F135 Engine for JSF



‘We found that the graphical user interface was easy to work with and made developing a rapid, intuitive test process a lot easier than creating it manually. This saving was further increased through the repeatability of tests utilising the automated regression testing facilities. This automated solution made our job a lot easier. The LDRA tool suite resulted in a saving of £2 million.’

Tom Roberts, Engineering Manager, Embedded Software and Systems, Ultra Electronics Dael

F-35 Lightning II



‘LDRA has proven they will support us in any way to get the job done especially in meeting demanding milestones. They provided outstanding support for several F-35 teammates: Lockheed Martin (Fort Worth), BAE (Warton), Northrop Grumman (El Segundo), Seaweed, and Honeywell which directly contributed to a successful first flight of the AA-1 aircraft. We continue to work closely with LDRA to develop the needed automated process support to ensure that our software meets program cost, schedule, and quality targets.’

John H. Robb, Air Vehicle Software Senior Manager, LMCO

Lexus, LS Sport, Sedan



‘LDRA has the ability to work with limited target hardware which is important in the automotive sector in order to meet the demands for cost reduction and downsizing. We use the LDRA tool suite as a benchmark for other third-party and similar software platform products.’

Akihito Iwai, Project Manager DENSO Japan

Space Shuttle Discovery



‘The tool’s reporting facilities for audit purposes are excellent. The team have also been able to dramatically increase the throughput of unit testing and as a result deliver the project on time.’

John Schneider, Director of Engineering, Neptec

Languages & Platforms

The **LDRA tool suite** is available for the following source code languages and host / target platforms:

Languages

Ada 83	Freescale Assemblers
Ada 95	Intel Assemblers
C	Texas Instruments Assemblers
C++	
Java	Languages shown in orange signify <i>TBrun</i> availability.

Host Platforms

Windows 9x/NT/2000/XP/Vista/Windows 7
 Unix, Linux
 HP OpenVMS (Alpha & IA-64 Itanium)

Target Platforms

IDE:		Processor:
Analog Devices	IAR	ARM
AONIX	iSYSTEM	Freescale
ARM	Keil	Infineon
Cosmic	QNX	Intel
Eclipse	TI	MIPS
Freescale	Renesas	PowerPC
GNU	TASKING	Renesas
Green Hills	Wind River	TI

Other languages, assemblers and host / target platforms are available. Please contact LDRA for more information.



Certificate Number: FM 28576

All brand names and product names mentioned herein are trademarks or registered trademarks of their respective companies. Picture acknowledgements: QinetiQ, Dräger, Lockheed Martin, Neptec, European Space Agency, Phillips Medical. Lexus Car picture, Copyright © 2005 - 2010 NihonCar.com Inc., NASA, Pratt & Whitney LDRA Ltd. reserves the right to change any specifications contained within this literature without prior notice.

Designed by Young Greenwood Design (01266) 226541

www.ldra.com

LDRA

LDRA Technology Inc. (US)

Lake Amir Office Park, 1250 Bayhill Drive
 Suite # 360, San Bruno CA 94066
 Tel: (650) 583 8880
 e-mail: info@ldra.com

LDRA UK & Worldwide

Portside, Monks Ferry,
 Wirral, CH41 5LH
 Tel: +44 (0)151 649 9300
 e-mail: info@ldra.com

LDRA
 LDRA Technology Inc.
 USA

LDRA
 LDRA Technology
 S.A.R.L. France