



Testing safety-critical software for the Def Stan 00-55 standard

Working with the MOD
to meet the challenges of
achieving software certification

www.ldra.com

Background to Def Stan 00-55?

It is a requirement that all safety-critical software supplied to the MOD must comply with the MOD standard Def Stan 00-55. The **LDRA tool suite** provides a powerful set of Static and Dynamic Analysis facilities which greatly assist in the process of ensuring that software applications meet this requirement.

Def Stan 00-55 Certification with the LDRA tool suite

This document describes the key software development and verification process requirements of the standard and how **LDRA's tool suite** can assist with meeting them.

The LDRA tool suite enables organisations to work towards:

- Static analysis (26.2.4)
- Dynamic testing (26.2.5)
- Testing and integration (37)
- Unit, integration & system testing (37.3)

Static Analysis (26.2.4)

Subset Analysis (26.2.41 b)

Most language variants of the **LDRA tool suite** enforce subsets of the specific languages. In the main these are safe subsets. All known constructs of the language which can give rise to problems are highlighted by means of an enforced set of configurable programming standards.

Metrics Analysis (26.2.41 b)

The **LDRA tool suite** produces all of the widely accepted software metrics relevant to safety issues.

Control Flow Analysis (26.2.41 c)

Control Flow Analysis is performed both on the program calling hierarchy and on individual procedures. The rules of structured programming are applied and defects reported. Checks are applied to ensure that the control flow patterns are predictable. Unreachable code is detected and highlighted by the tool.

Data Flow Analysis (26.2.41 d)

Data Flow Analysis is performed both at procedure level and as part of the system wide analysis, which is one aspect of integration testing.

Information Flow Analysis (26.2.41 e)

Information Flow Analysis is performed at both procedure level, as well as in system wide integration. The analysis gives the relationships of the input and output variables of procedures and programs, classifying the relationships into eight different categories. If predictions are available for these relationships they can be included in the source code in the form of comments (annotations) and the tool will then either confirm or refute these predictions.

Semantic Analysis (26.2.41 f)

Semantic Analysis is used to show that the code performs according to the definitions of the language constructs. Factors which affect these are compiler and chip defects, ambiguities and misunderstandings. The **LDRA tool suite** approaches this in a rigorous manner by means of Exact Semantic Analysis which is a Dynamic Analysis technique. Any form of static investigation must be at best approximate.

Safety Properties Analysis (26.2.41 g), & Object Code Verification (26.2.4.3)

In addition Static Analysis may also be performed on the decompiled object code to demonstrate that this possesses identical semantics to the source code.

Other Static Analysis checks are required in order to demonstrate that the compiler satisfies the language constraints.

Dynamic Analysis (26.2.4)

Dynamic Testing (26.2.5.1)

Dynamic Testing is the process of performing a formalised execution of the software with specific inputs and expected outcomes, with the execution sequences being monitored by tools in order to document the testedness.

The **LDRA tool suite** performs various forms of Dynamic Testing. It provides statement, branch and LCSAJ coverage. For those languages where it is appropriate various configurable forms of condition coverage, including MC/DC, are also available.

Additionally the results can be presented in the form of interprocedural call coverage or Dynamic Data Flow Analysis.

Exact Semantic Analysis is used to demonstrate that the code, as executed by a particular processor, behaves according to specific conditions or constraints. In Exact Semantic Analysis the annotations expressing these constraints can be refuted or confirmed over extensive patterns of control flow paths as enforced by the control flow metrics.

Dynamic Testing (26.2.5.2)

These tests should be conducted at unit, integration and system levels. The LDRA tool suite can produce Dynamic Testing results for all of these testing levels.

Metrics to support evidence of coverage are required, typically levels of testing at statement, branch and LCSAJ coverage levels. Additionally modified subcondition coverage (MC/DC) may be required.

In addition, dynamic checks of the language constructs are required in order to demonstrate that the compiler is predictable.

Use of Assembler (28.5)

Versions of the **LDRA tool suite** are available to test Assembler code.

Selection of Tools (29)

The **LDRA tool suite** has been utilised extensively in projects testing safety-critical applications. LDRA can produce documentation (of which this is a part) to justify tool selection. Further supporting evidence can be offered to demonstrate the stability, maturity, usability, interoperability and support of the tool suite.

Unreachable Code (30.2)

The **LDRA tool suite** detects unreachable code.

Coding Standards (36.1)

The **LDRA tool suite** can enforce coding standards that can be used to:

- Measure and monitor module size and inter-module data flow
- Monitor and document use of global variables and data, to help discourage their use
- Enforce strong typing
- Document use of any complex language features through structured programming verification

Object Code Verification (36.6)

A combination of high and low level language variants of the **LDRA tool suite** and disassemblers can be used to perform object code verification in specific environments. This facility is called DUAL.

Testing and Integration (37.1)

Dynamic and Performance Requirements (37.1.1)

The **LDRA tool suite** analysis enables dynamic and performance requirements to be met.

Testing Planning and Documentation (37.1.3)

The **LDRA tool suite** reports can be utilised in the planning and documentation of tests. LDRA tools can be utilised for white box and black box testing.

Regression Testing (37.1.6)

The **LDRA tool suite** features facilities to automatically perform Dynamic Analysis of regression tests. Documentation of regression tests can be performed and the LDRA tool suite analysis can be used to refine and enhance the regression process.

Testing Repeatability and Configuration Control (37.1.7)

All **LDRA tool suite** inputs, controls, command files and reports can be utilised within a configuration control system. They are easily utilised in a manner that ensures consistent repeatability.

Automated Comparison of Results (37.1.8)

The **LDRA tool suite** results are contained in a database that can be utilised by any documentation tool or configuration management tool for automatic comparisons.

Host/Host and Host/Target Testing (37.1.9)

The **LDRA tool suite** can be utilised in both Host/Host and Host/Target environments. Host/Target environments can include: real target hardware, hardware emulators, simulators etc.

Unit, Integration and System Testing (37.3)

LCSAJ Testing (37.3.1)

The **LDRA tool suite** can perform measurement of LCSAJ coverage.

Coverage Measures (37.3.2)

The **LDRA tool suite** supports all major coverage measurements.

Integration and System Testing (37.3.4)

The **LDRA tool suite** can be utilised in both integration and system testing.

Is the tool suite easy to use?

The tool suite's ease of use is a key issue when incorporating it into project procedures. The **LDRA tool suite** has been specifically enhanced to enable simple measurement of conformance to the various requirements of Def Stan 00-55. Reports are specifically tailored to give users Def Stan 00-55 information quickly and concisely, speeding up the testing procedure. Reports can be produced in either ASCII or HTML. Either format can be easily incorporated into a Word processor or DTP system. HTML has the added advantage of links and the ability to publish on the Internet or an intranet.

Proven Track Record in Def Stan 00-55 Certification

The **LDRA tool suite** is being utilised by companies around the world to meet Def Stan 00-55 standard. A cross-section of clients currently utilising the **LDRA tool suite** are:

Aeronautical Development Agency
Aerosystems International
Aerospatiale
BAE SYSTEMS
Bell Helicopters
CASA Madrid
Daimler-Benz Aerospace
Daimler-Benz AEG AGS
DLR
Dornier Freiderichshafen
Dunlop Aerospace
Eurocopter
Fiat Avio
GF-Sistemi Avionici
Honeywell
Hughes Aircraft Corporation
INTA

Logica Space & Communications
Lucas Aerospace
MTU Deutsche Aerospace
Raytheon
Rolls Royce
SENER
SFIM
SIA (Societa Italiano Avionici)
Smiths Aerospace
Steria
Thales Avionique
Teldix
TA Consultancy Services
TEC - Masters
TRW
Ultra Electronics
Westland Helicopters



LDRA Ltd. Portside, Monks Ferry,
Wirral, CH41 5LH, UK.
T: +44 (0)151 649 9300
F: +44 (0)151 649 9666
W: www.ldra.com
E: info@ldra.com

