Testing Avionics Software to DO-178B

Working with the avionics industry to meet the challenges of achieving certification economically

www.ldra.com

Background

In response to the increased use of software in airborne systems, the **Radio Technical Commission for Aeronautics** (RTCA) association, now known as RTCA Ltd., created the guidance document DO-178 "Software Considerations in Airborne Systems and Equipment Certification" which has come to be accepted as the international avionics certification standard for airborne software. Originally published in 1982, DO-178 has been revised several times to reflect the experience gained in the certification of avionic systems and in 1992 was completely rewritten and published as DO-178B.

The standard provides detailed guidelines for the production of all software for airborne systems and equipment (also known as aviation electronics, or avionics), whether it is safety critical or not. Prior to system development, a system safety assessment and hazard analysis are performed to determine the contribution of the system to potential failure conditions. The severity of failure conditions on the aircraft and its occupants are then used to determine a Software Level, per the table below:

Level	Failure Condition	Description
A	Catastrophic	Failure may cause a crash
B	Hazardous	Failure has a large negative impact on safety or performance, or reduces the ability of the crew to operate the plane due to physical distress or a higher workload, or causes serious or fatal injuries among the passengers.
C	Major	Failure is significant, but has a lesser impact than a Hazardous failure (for example, leads to passenger discomfort rather than injuries)
D	Minor	Failure is noticeable, but has a lesser impact than a Major failure (for example, causing passenger inconvenience or a routine flight plan change)
E	No effect	Failure has no impact on safety, aircraft operation, or crew workload

DO-178B translates these software levels into software specific objectives that must be satisfied during the development process. An assigned software level determines the level of effort required to show compliance with certification requirements, which varies with the failure condition category. This means that the effort and expense of producing a system critical to the continued safe operation of an aircraft (e.g. a fly-by-wire system) is necessarily higher than that required to produce a system with only a minor impact on the aircraft in the case of a failure (e.g. the in-flight entertainment system).

Intended to provide confidence in the safety of avionic systems by ensuring that they comply with airworthiness objectives, DO-178B covers the complete software lifecycle: planning, development and integral processes to ensure correctness, control and confidence in the software. These integral processes include requirements traceability, software design, coding and software validation and verification.

To achieve the airworthiness objectives for avionic systems, both software analysis and requirements traceability tools are essential for cost conscious avionics projects. LDRA has extensive experience in this specialised area with the ***LDRA tool suite®*** which provides the most comprehensive requirements traceability, source code analysis and testing facilities for assisting companies to meet DO-178B software development and verification requirements. As the application of DO-178B becomes more widespread it is essential that the choice of tools is based on known expertise, especially when it comes to emerging issues, such as software security. The ***LDRA tool suite*** was the first tool to be utilised for certification to the DO-178B required standard for airborne systems as well as its companion standard, DO-278 for ground-based systems.

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.

DO-178B Process Objectives

DO-178B recognises that software safety and security must be addressed in a systematic way throughout the software development life (SDLC). This includes the requirements traceability, software design, coding, validation and verification processes used to ensure correctness, control and confidence in the software.

Key elements of the DO-178B SDLC are the practices of traceability and coverage analysis. Traceability (or Requirements Traceability) refers to the ability to link system requirements to software high-level requirements, from software high-level requirements to low-level requirements, and then from low-level requirements to source code and the associated test cases. Coverage (or code coverage analysis) refers to measuring the degree to which the source code of a system has been tested. Through the use of these practices it is possible to ensure that code has been implemented to address every system requirement and that the implemented code has been tested to completeness.

There are 2 sections of the DO-178B guidance document where the use of the ***LDRA tool suite*** offers significant benefits:

- Section 5.0 - Software Development Processes
- Section 6.0 - Software Verification Process

Software Development Processes (Section 5.0)

Four high level activities are identified in the DO-178B Software Development Processes section; Software requirements process, Software design process, Software coding process and Integration process. In addition, there is a section on requirements traceability (section 5.5) that embodies the evolution and traceability of requirements from system level requirements to source code. The ***LDRA tool suite*** offers significant benefits when used in conjunction with Section 5.3, the Software Development Processes and Section 5.5, the Traceability section.

As part of Section 5.3, the software development process, DO-178B specifies that software must meet certain software coding process requirements. These include adherence to a set of software coding standards and traceability from low level design requirements to the source code and object code.

Further definition of the software coding standards are provided in Section 11.8 of DO-178B:

- Programming language(s) to be used and/or defined subset(s). For a programming language,reference the data that unambiguously defines the syntax, the control behaviour, the data behaviour and side-effects of the language. This may require limiting the use of some features of a language.
- Source code presentation standards, for example, line length restriction, Indentation, and blank line usage and source code documentation standards, for example, name of author, revision history, inputs and outputs, and affected global data.
- Naming conventions for components, subprograms, variables and constants.
- Conditions and constraints imposed on permitted coding conventions, such as the degree of coupling between software components and the complexity of logical or numerical expressions and rationale for their use.
- Constraints on the use of the coding tools.

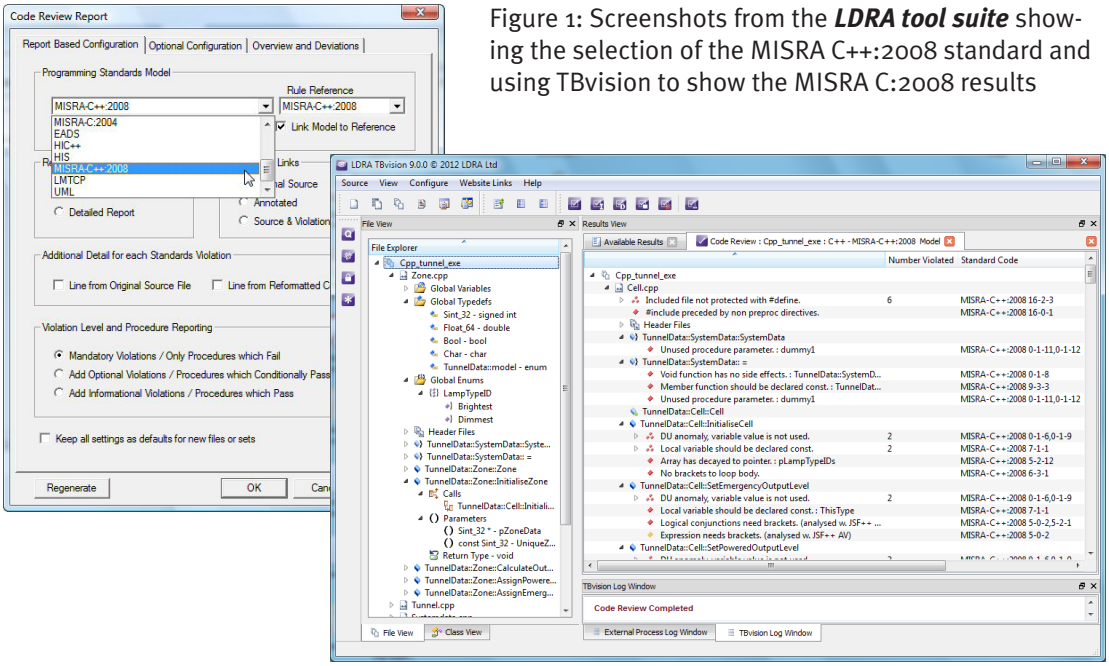

Figure 1: Screenshots from the ***LDRA tool suite*** showing the selection of the MISRA C++:2008 standard and using TBvision to show the MISRA C:2008 results

Using the latest source code analysis technology, the ***LDRA tool suite*** provides code analysis features to enable software coding standards compliance and detailed low level source code documentation. This makes compliance checking easier, less error prone and more cost effective than the manual equivalent.

In section 5.5, DO-178B mandates that the correctness of the requirements-based development and verification process is determined by requirements coverage or traceability. This analysis assures that software requirements are properly associated with the requisite test cases and can be traced from their highest level through the design to the final implementation and deployment of the software on the hardware (or target).

The ***LDRA tool suite*** offers a requirements traceability/coverage tool that is integrated with LDRA's code review, data and control coupling and code coverage tools. The integration of these tools offers unparalleled support for DO-178B certification. This makes the creation, management, maintenance and documentation of the requirements traceability matrix through disparate requirements documents and down to the source code and test cases much more straightforward and cost effective than the manual equivalent.

Software Verification Process (Section 6.0)

Per Section 6.1 of DO-178B, the objectives of the Software Verification Process are "*to detect and report errors that may have been introduced during the software development processes.*" Per Section 6.2 of DO-178B, these objectives are satisfied "*through a combination of reviews, analyses, the development of test cases and procedures, and the subsequent execution of those test procedures. Review and analyses provide an assessment of the accuracy, completeness and verifiability of the software requirements, software architecture and source code.*" The ***LDRA tool suite*** is of particular value throughout both the review and the analysis portions of this process.

This section describes the use of the ***LDRA tool suite*** during the traceability and analysis process, while the next section describes the use of the tool suite during the structural coverage analysis process described in Section 6.4.4.2b of DO-178B.

One of the key elements of the review process described in Section 6.3 of DO-178B is ensuring that requirements are traceable from the system requirements through to the source code used to implement them. The ***TBreq®*** requirements traceability/coverage tool from LDRA can be used throughout this process to document the evolution of system requirements through to source code, including derived requirements. To assist in the documentation required as an artifact of this process, this tool will also generate a Requirements Traceability Matrix.

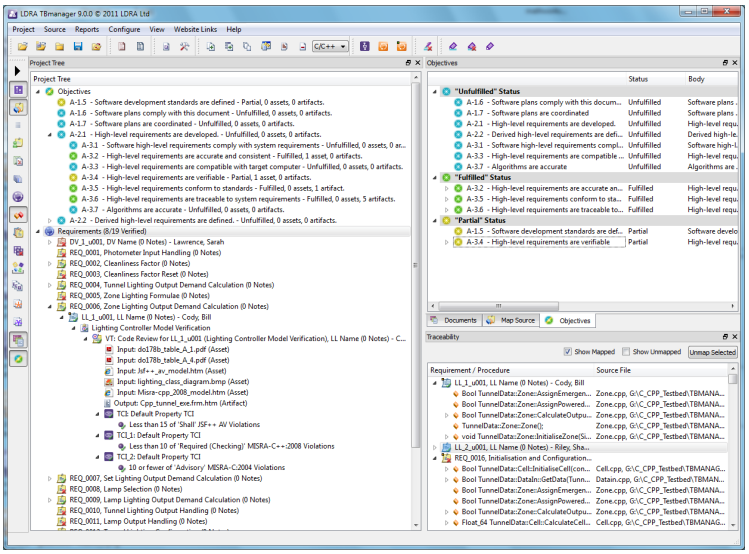

Figure 2: ***TBmanager*** graphical user interface for tracing requirements

Specific to the review and analysis of the source code itself, Section 6.4.3d requires that the source code be verified for compliance with the software code standards, including ensuring that code complexity does not exceed a level consistent with the system safety objectives. In addition, Section 6.4.3f requires that the source code be verified for accuracy and consistency ensuring correctness when any of the following are used:

- Uninitialised variables or constants.
- Unused variables or constants.

The ***LDRA tool suite*** makes compliance checking easier, less error prone and more cost effective by providing the tools for assessing the code under review against a software code standard, highlighting any areas of non-conformance. In addition, the ***LDRA tool suite*** can assess the complexity of the code under review to ensure that it stays below a safe threshold for the system. Furthermore, the data flow analysis capabilities of the ***LDRA tool suite*** can be used to identify any uninitialised and unused variables or constants.

DO-178B Structural Coverage Analysis Objectives

Structural Coverage Analysis (SCA) is used to analyse the effectiveness of the requirements-based test procedures. All testing must be performed at the system level and be driven from the software requirements - that is, using requirements-based functional tests. Following that, SCA is applied to measure the effectiveness of this testing, i.e. measuring how much of the code has been exercised as a result of the requirements-based functional tests. The feedback gained as a result of this analysis helps to validate that the code base has been tested to completeness, and also to ensure that there is no unintended code in the system software by validating that all of the code is traceable to a specific system requirement or set of requirements.

SCA is one of the closing steps in the requirements coverage/traceability process. SCA is an empirical measurement of requirements test effectiveness, helping to link the source code implementation and associated test procedures back to the system requirements while ensuring that the system code has been tested to the level of completeness required by the system software level.

The ***LDRA tool suite*** is the only product that may be qualified for use as a DO-178B Software Verification Tool that provides an independent analysis of structural coverage from the high level software requirement through low level requirements, the design, to the source code and down to the object code.

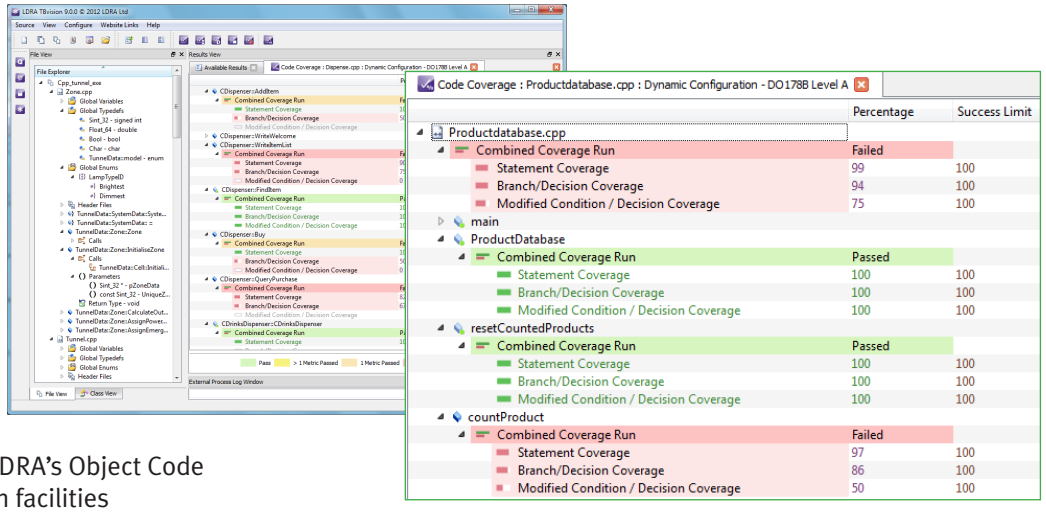

Figure 3: LDRA's Object Code Verification facilities

DO-178B imposes very strict structural coverage analysis requirements on the software because if code coverage is not monitored, there is the possibility that errors will still be present in code that has not been executed by any of the test data. Through automatic source code instrumentation, the ***LDRA tool suite*** makes SCA analysis easier, less error prone, and more cost effective by providing the tools for measuring the areas of code that are executed at run time and reporting on the code that has and has not been executed, facilitating rapid identification of missing or inadequate test data.

Through the ***LDRA tool suite's*** measurement of these coverage metrics, testing strategies can be implemented and enhanced to meet the required degree of coverage appropriate to the safety level of the software. This will greatly increase confidence in the tested code.

Following is the pertinent extract from DO-178B with respect to SCA:

6.4.4.2 Structural Coverage Analysis

The objective of this analysis is to determine which code structure was not exercised by the requirements-based test procedures. The requirements based test cases may not have completely exercised the code structure, so structural coverage analysis is performed and additional verification produced to structural coverage. Guidance includes:

a. *The analysis should confirm the degree of structural coverage appropriate to the software level.*

b. *The structural coverage analysis may be performed on the source code, unless the software is level A and the compiler generates object code that is not directly traceable to source code statements. Then, additional verification should be performed on the object code to establish the correctness of such generated code sequences. A compiler-generated array bound check in the object code is an example of object code that is not directly traceable to the source code.*

c. *The analysis should confirm data coupling and control coupling between the code components.*

The SCA objectives for each software level are summarised in the following table

Item	Description	DO-178B Reference	DO-178B Level A	DO-178B Level B	DO-178B Level C	DO-178B Level D
5	Test coverage of software structure (MC/DC) is achieved	6.4.4.2	✓	Not Required	Not Required	Not Required
6	Test coverage of software structure (decision coverage) is satisfied	6.4.4.2a 6.4.4.2b	✓	✓	Not Required	Not Required
7	Test coverage of software structure (statement coverage) is satisfied	6.4.4.2a 6.4.4.2b	✓	✓	✓	Not Required
8	Test coverage of software structure (data coupling and control coupling) is achieved	6.4.4.2c	✓	✓	✓	Not Required
Note: Items 5, 6, 7 and 8 are not required for DO-178B Levels D and E. Items 1 to 4 (not shown) are manual procedures.						

✓ - Satisfied by the ***LDRA tool suite***, which can be used to satisfy the 'with Independence' requirement

In addition to the DO-178B specific coverage requirements above, the ***LDRA tool suite*** also provides further industry standard coverage measures as follows:

- LCSAJ Coverage (one of the most effective test analyses)
- Branch Condition Coverage (BCC)
- Branch Condition Combination Coverage (BCCC)

Although not strictly SCA, DO-178B Section 6.4.42c requires that analysis be performed to confirm the data and control coupling between the code components under test. The following are details of the specific facilities offered by the ***LDRA tool suite***, which can be used to satisfy this DO-178B requirement.

Control Coupling

Defined by DO-178B to be "The manner or degree by which one software component influences the execution of another software component." The ***LDRA tool suite*** provides a visual representation of the control coupling dependence of a given software component on those components that call it, or are called by it, including calling frequency. This information may also be mapped back directly to the source code by 'drilling-down' to the specific predicates within the source code which must be satisfied in order to affect the call.

In addition to the Static Analysis facilities described above, in the dynamic domain the Dynamic Callgraph Display will demonstrate the degree to which the identified control coupling has been exercised at run-time.

Data Coupling

Defined by DO-178B to be "The dependence of a software component on data not exclusively under the control of that software component," the ***LDRA tool suite*** provides data coupling information in both the Static and Dynamic analysis domains, showing ALL instances of the data items accessed by a software component. This includes local variables declared within the scope of the component and global variables accessed by the component, but declared elsewhere. Significantly the ***LDRA tool suite*** tracks and reports these data items across file and procedure boundaries even in cases where they are aliased as parameters to procedure calls.

In the Dynamic domain the Dynamic Data Flow Coverage facility provided by the ***LDRA tool suite*** indicates which data components have been accessed at run-time providing the data coupling for a particular test case.

Tool Selection

The use of traceability and analysis tools for an avionics project that must meet the DO-178B certification requirements offers significant productivity and cost benefits. Tools make compliance checking easier, less error prone and more cost effective. In addition, they make the creation, management, maintenance and documentation of requirements traceability straightforward and cost effective. When selecting a tool to assist in achieving DO-178B acceptance the following criteria should be considered:

- Does the tool provide a complete 'end-to-end' Requirements Traceability capability to enable linkage and documentation from all levels to the source code and associated test cases?
- Does the tool enable analysis for all Structural Coverage Analysis requirements as laid out in section 6.4.4.2 of the standard?
- Can the tool perform MC/DC analysis in assignment statements as well as conditional statements?
- Is there tool availability for all the languages required in your project?
- Has the tool been utilised in this manner successfully already?
- Will the tool vendor assist in tool qualification?
- Is tool support both flexible and extensive enough to meet changing requirements?
- Is the tool easy to use?

The *LDRA tool suite* meets all of the above criteria.

Availability

For availability information regarding the ***LDRA tool suite***, please refer to the LDRA website, or contact LDRA.

Proven Track Record in DO-178B Qualification

The ***LDRA tool suite*** is being utilised by companies around the world to meet DO-178B and other avionics standards. For a cross section of LDRA customers please refer to the Client Base List available on our website.

Tool Qualification

Certification authorities such as the FAA, CAA and JAA undertake tool qualification on a project by project basis. This means that, when considering tools, assistance in tool qualification is essential. As an integral part of the LDRA DO-178B Analysis Package, LDRA offers a Tool Qualification Support Pack and agrees to enable clients and certification authorities to audit the ***LDRA tool suite*** for use in your project. This audit process has already been undertaken by many existing LDRA customers.

Flexible Tool Support

As certification authorities gain further experience in applying DO-178B to projects, customers must be concerned that their tool vendors have the flexibility to cope with changing requirements. For example the FAA is currently requesting coverage information on combinations of constructs that only the ***LDRA tool suite*** can measure. In addition DO-178B may require coverage of implied XOR in the C language. LDRA is committed to helping existing customers meet changing requirements now and in the future.

Ease of Use

The tool's ease of use is a key issue when establishing project procedures. The ***LDRA tool suite*** has been specifically developed to enable simple measurement of conformance to the various levels of DO-178B. Reports are tailored to give users DO-178B information quickly and concisely, speeding up the testing procedure. Reports can be produced in either ASCII or HTML formats. Either format can be easily incorporated into a word processor or DTP system. HTML has the added advantage of links and the ability to be published on an intranet.

The *LDRA tool suite* is utilised on the following avionics projects:

- Airbus A320/A330/A340/A380
- Boeing 777/787
- C-5 RERP
- De Haviland DASH-8
- EH101 Merlin
- Eurocopter
- EuroFighter Typhoon
- Eurojet
- Extended Air Defence Testbed EADTB
- F/A-22 Raptor
- F-35 Lightning II
- F-16 Fighting Falcon
- HAL Light Attack Aircraft
- Hawk
- Hüygens Satellite
- Nimrod 2000
- Short Range UAV
- T-50 Golden Eagle
- Tornado

Certificate Number FM 26376

www.ldra.com

LDRA

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

LDRA Technology, Inc.
2540 King Arthur Blvd, Suite #228 Lewisville Texas 75056
Tel: +1 (855) 855 5372
e-mail: info@ldra.com

LDRA Technology Pvt. Ltd
#2989/1B, 3rd Floor, 12th Main, 80 Feet Road,
HAL II Stage, Bangalore- 560008. Near BSNL Building
Tel: +91 80 4080 8707
e-mail: india@ldra.com

DO-178B v3.3 08/13