# Example TQSP

# Table of Content

# Example of Code Review TQSP

# PRC TAC Document

LDRA Software Technology

RTCA/DO-178B

Programming Rules Checking (PRC)

Tool Accomplishment Summary

for the

C/C++ LDRA tool suite

Release Version 9.1.1

# PRC TAC Document (Continued)

**Table of contents**

# PRC TAC Document (Continued)

### 4.1 Test Results Summary

**Table 4: Rules Disabled / Applied (SCSR-01 and SCSR-02)**

| LDRA Rule ID (Standard.html) | Rule Description | Test Case File | Applied | Pass | Fail |
|---|---|---|---|---|---|
| XX | XX Description | XX_Test_Case.c | | | |
| YY | YY Description | YY_Test_Case.c | | | |
| ZZ | ZZ Description | ZZ_Test_Case.c | | | |

**To the applicant**: In the above table you should record the programming rule checks that were disabled in order to verify SCSR-02, the test cases that were applied in order to verify these actions and the test outcomes of these test cases. The 'Applied' column may be used to indicate (Y or N) whether or not the tool suite correctly overrode the disabling of the associated programming rule in accordance with SCSR-01.

**Table 5: [redacted] Test Results Summary**

| LDRA Rule ID (Standard.html) | Rule Description | Test Case File | Pass | Fail |
|---|---|---|---|---|
| 1 S | Procedure name reused. | Static_001.c | | |
| 2 S | Label name reused. | Static_002.c | | |
| 4 S | Procedure exceeds *** reformatted lines. | Static_004.c | | |
| 5 S | Empty then clause. | Static_005.c | | |
| 6 S | Procedure pointer declared. | Static_006.c | | |
| 7 S | Jump out of procedure. | Static_007.c | | |

# PRC TOR Document

LDRA Software Technology

RTCA/DO-178B

Programming Rules Checking (PRC)

Tool Operational Requirements

for the

C/C++ LDRA tool suite

Release Version 9.1.1

# PRC TOR Document (Continued)

**Table of contents**

# PRC TOR Document (Continued)

## 3 Tool Operational Requirements

### 3.1 Application of Specified Coding Standard and Rules

**[SCSR-01]** If the coding standard model is selected to be applied as a whole, LDRA Testbed *shall* only apply, and report against, the subset of the available programming rules checks indicated in section 3.1.1 of this TOR.

**[SCSR-02]** LDRA Testbed *shall* apply all of the programming standard rule checks indicated as applicable in section 3.1.1 of this TOR irrespective of whether any of these checks have been disabled in the CPEN.DAT file that is being applied at the time of the analysis.

The selection of additional rules, external to the selected coding standard, is addressed in section 3.1.2 of this TOR.

#### 3.1.1 Coding Standard Rules

**[CSR-01]** LDRA Testbed *shall* apply each [redacted] rule indicated below:

| LDRA Rule ID (Standard.html) | Rule Description | Test Case File |
|---|---|---|
| 1 S | Procedure name reused. | Static_001.c |
| 2 S | Label name reused. | Static_002.c |
| 4 S | Procedure exceeds *** reformatted lines. | Static_004.c |
| 5 S | Empty then clause. | Static_005.c |
| 6 S | Procedure pointer declared. | Static_006.c |
| 7 S | Jump out of procedure. | Static_007.c |
| 8 S | Empty else clause. | Static_008.c |
| 9 S | Assignment operation in expression. | Static_009.c |
| 11 S | No brackets to loop body (added by Testbed). | Static_011.c |
| 12 S | No brackets to then/else (added by Testbed). | Static_012.c |
| 13 S | goto detected. | Static_013.c |
| 20 S | Parameter not declared explicitly. | Static_020.c |

# PRC TQP Document

LDRA Software Technology

RTCA/DO-178B

Programming Rules Checking (PRC)

Tool Qualification Plan

for the

C/C++ LDRA tool suite

Release Version 9.1.1

# PRC TQP Document (Continued)

**Table of Contents**

# PRC TQP Document (Continued)

## 6 Activities to be Performed

The following sections identify the activities to be performed by the applicant in qualifying LDRA Testbed. These activities should be performed in the order presented, and should be accomplished in the general time frame as indicated for each step.

It is assumed that the editing/creation of data identified hereafter will be performed in accordance with the qualifying project's software plans, including the performance of internal reviews, archive of review data, and approval/release of data into the appropriate configuration management environment.

### 6.1 Tool Qualification Planning

The tool qualification planning activities include:

- Customisation of this TQP with project-specific information (as identified in the TVP for LDRA Testbed).
- Completion of the project PSAC, with specific details of the certification credit sought by the applicant and details of the exact version of the LDRA tool suite in use.
    - The LDRA provided TOR should be reviewed to ensure that the tool operational requirements are commensurate with the needs of the software verification process.
- Submittal of the PSAC and TQP to the certification authority for their review and approval.

Tool qualification planning should be performed in conjunction with the project planning phase. Selection of the software development and verification environments is an important part of the project planning tasks under RTCA/DO-178B.

### 6.2 Tool Verification

The tool verification phase activities include:

- Installation and configuration of the LDRA Testbed in the applicant's environment.
- Informal validation that the installed tool is functional.
- Execution of the LDRA provided qualification test suite.
    - Sections of the test suite that cover tool capabilities not under qualification consideration by the applicant (such as rules deselected from the applied coding standard) may be omitted.
    - Additional operational requirements and associated test cases may be required for extensions of the selected coding standard.
- Verification that the actual results of the applicant-run test suite match the expected results.
    - This may be accomplished by detailed review and comparison of the actual test results against the LDRA provided expected results.
    - Discrepancies between the actual and expected results are identified and dispositioned in the project's TAS.
- Archiving of the actual test results as CC2 data.

Tool verification should be accomplished prior to commencement of formal testing of the airborne software. Completion of all tool verification activities prior to starting formal test execution ensures that the qualification of the tool will be considered valid for gaining the certification credit being sought. At the latest, the tool verification phase should be complete prior to any certification authority verification review (SOI-3).

# PRC TQP Document (Continued)

## 7 Tool Qualification Data to be Produced

The following tool qualification data will be produced or otherwise used under this TQP:

| Item | ID | Source | Description | CC | Submit |
|---|---|---|---|---|---|
| 1 | PSAC | Project | The Plan for Software Aspects of Certification is the primary means used by the certification authority for determining whether an applicant is proposing a software life cycle that is commensurate with the rigour required for the level of software being developed. The PSAC is required to contain specific information regarding the intent to qualify the LDRA Testbed, and should reference this TQP. The PSAC should identify the specific certification credit sought, which may be identified by reference to specific sections in the TQP. | CC1 | Yes |
| 2 | TQP | LDRA① | This LDRA provided Tool Qualification Plan is customised by the applicant in accordance with the instructions in the TVP for LDRA Testbed. This TQP contains a description of the tool and its architecture, details of the certification credit sought, identifies the tool qualification activities to be performed, and summarises the tool qualification data to be produced. | CC1 | Yes |
| 3 | TOR | LDRA | The LDRA Tool Operational Requirements identifies the operational requirements eligible for qualification under this TQP. The applicant should identify the specific requirements being qualified in the TAS and SAS. As the TOR is referenced by submitted data, it should be controlled as CC1 data by the applicant. | CC1 | Avail. |
| 4 | TVP | LDRA | The Tool Verification Plan is an LDRA provided instructional script for verification of LDRA Testbed. The TVP refers to the TVCP to identify the test cases associated with each TOR. The applicant should be prepared to provide this document to the certification authority upon request. | CC2 | Avail. |

# PRC TQP Document (Continued)

| | | | | | |
|---|---|---|---|---|---|
| 5 | TVCP | LDRA | The LDRA provided Tool Verification Cases and Procedures include all of the source code, test cases to be executed against the source code, and expected results for use in verifying the applicant's installation of the LDRA Testbed. The applicant is responsible for ensuring that the test cases contained in the TVCP are sufficient to cover all of the coding rules in use on the project. | CC2 | Avail. |
| 6 | TVR | Project② | The Tool Verification Results are the actual results generated by executing the TVCP against the test source code in the applicant's environment. These actual results are compared against the expected results contained in the TVCP to show that the tool is operating properly under normal operating conditions. | CC2 | Avail. |

| | | | | CC | Submit |
|---|---|---|---|---|---|
| 7 | TAS | LDRA① | The LDRA provided Tool Accomplishment Summary is customised by the applicant in accordance with the instructions contained in the TVP to summarise the results of the tool verification, thereby providing the certification authority with the data required to accept the qualification of LDRA Testbed within the project's defined process. This document is the primary data for showing correct operation of the qualified tool under normal operating conditions. | CC1 | Yes |
| 8 | SAS | Project | The Software Accomplishment Summary is the primary data for showing compliance with the Plan for Software Aspects of Certification. The SAS summarises the results of the tool qualification, and references the TAS. | CC1 | Yes |
| 9 | SECI | Project | The Software Life Cycle Environment Configuration Index identifies qualified tools and their associated tool qualification data. | CC1 | Avail. |

① The LDRA provided template is customised by the qualifying project.

② The TVR is produced by execution of the LDRA provided test suite in the project environment.

# Code Review Report

## LDRA Testbed ® Code Review Report

**File :** [redacted]\static_001.c

**Overall Result: FAIL**

| Report Production | Report Configuration | Analysis phases |
|---|---|---|
| • C/C++ LDRA Testbed Version: 9.4.3<br>• Produced On: Fri Feb 07 2014 at 14:15:46<br>• Generated from GLH Repository | • Report Level: Summary Report<br>• Procedures Reported: All Procedures<br>• Programming Standards Model: MISRA-C:2012<br>• Line Numbers refer to: Original Source File<br>• Violation Details: Violations Only<br>• Reporting Scope: Source file and associated header | • Static: Yes<br>• Complexity: Yes<br>• Static Data Flow: Yes<br>• Information Flow: No<br>• Cross Reference: Yes |

its

# Code Review Report (Continued)

## Overall Code Review Summary

**Totals for Violated Code Review Standards**

| Number of Violations | LDRA Code | (M) Mandatory Standards | MISRA-C:2012 Code |
|---|---|---|---|

No Mandatory Standards Violated

| Number of Violations | LDRA Code | Required Standards | MISRA-C:2012 Code |
|---|---|---|---|
| 1 | 1 S | Procedure name reused. | MISRA-C:2012 R.5.8,R.5.9 |

| Number of Violations | LDRA Code | Advisory Standards | MISRA-C:2012 Code |
|---|---|---|---|

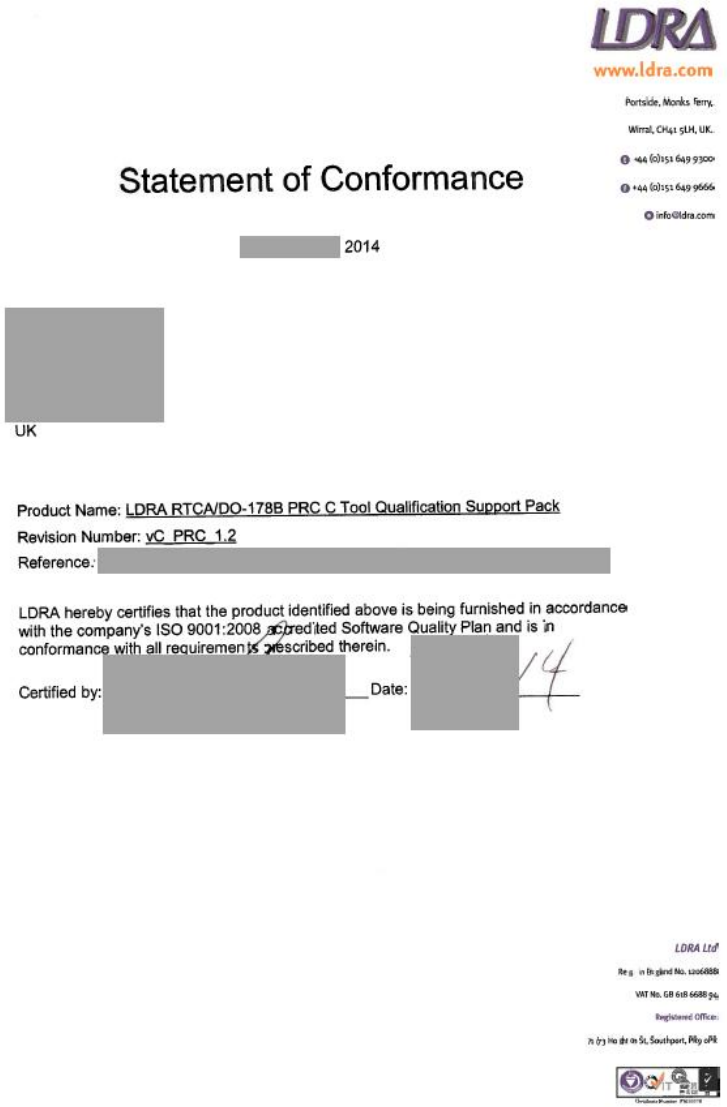
No Advisory Standards Violated

| Number of Violations | LDRA Code | Document Standards | MISRA-C:2012 Code |
|---|---|---|---|

No Document Standards Violated

| | |
|---|---|
| Number of Mandatory Standards checked | 18 |
| Number of Required Standards checked | 280 |
| Number of Advisory Standards checked | 35 |
| Number of Document Standards checked | 3 |
| Total Standards checked | 333 |
| Total Standards checked including Document | 336 |

# Statement Of Conformance (SoC)



LDRA

www.ldra.com

Portside, Monks Ferry,
Wirral, CH41 5LH, UK.
+44 (0)151 649 9300
+44 (0)151 649 9666
info@ldra.com

## Statement of Conformance

[redacted] 2014

[redacted]
UK

Product Name: LDRA RTCA/DO-178B PRC C Tool Qualification Support Pack
Revision Number: vC_PRC_1.2
Reference: [redacted]

LDRA hereby certifies that the product identified above is being furnished in accordance with the company's ISO 9001:2008 accredited Software Quality Plan and is in conformance with all requirements prescribed therein.

Certified by: [redacted] Date: [redacted] /14

LDRA Ltd
Registered in England No. 1206888
VAT No. GB 618 6688 94
Registered Office:
71-73 Hoghton St, Southport, PR9 0PR

# Example of Code Coverage TQSP

# SCA TAS Document

LDRA Software Technology

IEC 62304

Structural Coverage Analysis (SCA)

Tool Accomplishment Summary

for the

C/C++ LDRA tool suite

Release Version 9.2.0

# SCA TAS Document(Continued)

**Table of contents**

# SCA TAS Document(Continued)

## 4.1 Test Results Summary

**Table 4: Test Results Summary**

| Test File | Description | Pass | Fail |
|---|---|---|---|
| *Test Case Set:* ***SCA_C_1_2_3_IFS*** | | | |
| IF_THEN.C | This test example is intended to demonstrate 100% SCA (of the selected coverage criteria) of a simple IF-THEN statement. | | |
| IF_THEN_ELSE_IF_THEN_ELSE.C | This test example is intended to demonstrate 100% SCA (of the selected coverage criteria) of an IF-THEN-ELSE with an IF-THEN-ELSE in the ELSE clause. | | |
| IF_THEN_ELSE_WITH_2COND_ASSIGNED_AND_OR.C | This test example is intended to demonstrate 100% SCA (of the selected coverage criteria) of an IF-THEN-ELSE which evaluates the result of an assignment of a mixed BOOLEAN && (AND) and \|\| (OR) expression with two sub-conditions. | | |
| IF_THEN_ELSE_WITH_2COND_ASSIGNED_OR_AND.C | This test example is intended to demonstrate 100% SCA (of the selected coverage criteria) of an IF-THEN-ELSE which evaluates the result of an assignment of a mixed BOOLEAN \|\| (OR) and && (AND) expression with two sub-conditions. | | |
| IF_THEN_ELSE_WITH_2NEGATED_AND.C | This test example is intended to demonstrate 100% SCA (of the selected coverage criteria) of an IF-THEN-ELSE with a BOOLEAN && (AND) control expression in which both of the two conditions are negated. | | |
| IF_THEN_ELSE_WITH_3COND_ASSIGNED_AND.C | This test example is intended to demonstrate 100% SCA (of the selected coverage criteria) of an IF-THEN-ELSE which evaluates the result of an assignment of a BOOLEAN && (AND) expression with three sub-conditions. | | |
| IF_THEN_ELSE_WITH_3COND_ASSIGNED_OR.C | This test example is intended to demonstrate 100% SCA (of the selected coverage criteria) of an IF-THEN-ELSE which evaluates the result of an assignment of a BOOLEAN \|\| (OR) expression with three sub-conditions. | | |
| IF_THEN_ELSE_WITH_AND.C | This test example is intended to demonstrate 100% SCA (of the selected coverage criteria) of an IF-THEN-ELSE with a BOOLEAN && (AND) control expression. | | |
| IF_THEN_ELSE_WITH_ASSIGNED_AND.C | This test example is intended to demonstrate 100% SCA (of the selected coverage criteria) of an IF- | | |

# Code Coverage Report

## LDRA Testbed ® Dynamic Coverage Analysis Report

File : [redacted]

Overall Result (For File): Coverage Metrics required to achieve DO-178C Level A Attained

Statement (TER1) = 100 % Branch/Decision (TER2) = 100 % MC/DC : Not Applicable

## LDRA Testbed ® Dynamic Coverage Analysis Report

File : Fri Feb 07 2014 at 13:57:58 by LDRA Testbed : Version 9.4.3

Overall Result (For File): Coverage Metrics required to achieve DO-178C Level A Attained

Statement (TER1) = 100 % Branch/Decision (TER2) = 100 % MC/DC : Not Applicable

# Code Coverage Report (Continued)

**Coverage Metrics required to achieve DO-178C Level A Attained**

Statement (TER1) = 100 % Branch/Decision (TER2) = 100 % MC/DC : Not Applicable

**Statement Coverage Profile**

```
LINE NUMBER                                                                                    PREVIOUS  CURRENT
REF.  (SOURCE)     STATEMENT                                                                   RUNS      RUN       COMBINED
31 (29)          void                                                                          -         -         -
32                 f_if1 (                                                                     2         2         4
33                 int f_if1_input1 )                                                          -         -         -
34 (30)            {                                                                           -         -         -
35 (31)              int                                                                       -         -         -
36                     f_if1_local1 = 0 ;                                                      -         -         -
37 (33)              /*************************************************************          -         -         -
38 (34)                TEST DATA REQUIREMENTS (LEVEL-A or B)                                 -         -         -
39 (36)                   f_if1_input1 = 0                                                   -         -         -
40 (37)                   f_if1_input1 = 1                                                   -         -         -
41 (39)              *************************************************************/         -         -         -
42 (41)              /*************************************************************          -         -         -
43 (42)                TEST DATA REQUIREMENTS (LEVEL-C)                                      -         -         -
44 (44)                   f_if1_input1 = 0                                                   -         -         -
45 (46)              *************************************************************/         -         -         -
46 (48)              if                                                                      2         2         4
47                     (                                                                     2         2         4
48                     f_if1_input1 == 0                                                     2         2         4
49                     )                                                                     2         2         4
50                     {                                                                     1         1         2
51                       f_if1_local1 = 1 ;                                                  1         1         2
52                     }                                                                     1         1         2
53 (49)            }                                                                         2         2         4
```

| Summary | Prev. Runs | Current Run | Combined |
|---|---|---|---|
| **Number of Executable Lines** | 9 | | |
| Number Executed | 9 | 9 | 9 |
| Number not Executed | 0 | 0 | 0 |
| **Statement Coverage (%)** | **100** | **100** | **100** |

# Test Manager Report

## LDRA Testbed ® Test Manager Report

**File :** [redacted]\if_then.c

| Report Production | Report Configuration |
|---|---|
| • C/C++ LDRA Testbed Version 9.4.3 | • Report Format: Procedure Listing<br>• Reporting Scope: Source file and associated header |

### TBrun Unit / Module Test

| Name of Sequence | Test Cases / Mode | Regression Analysis | Suspension Information |
|---|---|---|---|
| s1 | 2 - White | 2 Pass | No Suspensions |

### Test Verification

| Procedure | Statement | Branch | MC/DC |
|---|---|---|---|
| f_if1 | 100 | 100 | [ No BCs ] |

| Top of Report |

End of Test Manager Report