GNATcoverage DO-178B/ED-12B Qualification Material: Plans

Release 0.8

AdaCore for Thales ADIRU

July 07, 2011

This page is intentionally left blank.

CONTENTS

This page is intentionally left blank.

Table 1: Authors

Name	Company	Email
Matteo Bordin	AdaCore	bordin@adacore.com
Olivier Hainque	AdaCore	hainque@adacore.com

Table 2: Revised by

Name	Company	Email
Cyrille Comar	AdaCore	comar@adacore.com

This page is intentionally left blank.

CHAPTER

ONE

DOCUMENTATION INTRODUCTION

This section contains an introduction to qualification material for GNATcoverage.

1.1 Document Purpose

The purpose of this document is to describe the applicable processes to qualify GNATcoverage; it also assigns specific activities to each interested party.

1.2 Definitions List

Environment

The context on which GNATcoverage is used.

GNATcoverage

A tool performing structural coverage analysis of Ada programs.

Qualification environment

The environment in which GNATcoverage is qualified.

Report

A file containing the GNATcoverage output.

Test

A test is an concretization of a testcase with precise values and parameters.

Testcase

A part of the testing strategy used to verify a given Tool Operational Requirement.

Tool Operational Requirement (TOR)

A TOR describes the expected behaviour of a tool from the point of view of the user.

User environment

The environment in which GNATcoverage is used.

1.3 Referenced Documents

AE09

Bordin et al.: *Couverture: An Innovative Open Framework for Coverage Analysis of Safety Critical Applications* - Ada User Journal, December 2009.

DO-178B and ED-12B

EUROCAE : SOFTWARE CONSIDERATIONS IN AIRBORNE SYSTEMS AND EQUIPMENT CERTIFICATION

ERTS2010

Bordin et al: *Couverture: An Innovative and Open Coverage Analysis Framework for Safety-Critical Applications* - ERTS2 2010

GNAT Pro UG

AdaCore: GNAT Pro User Guide. Available as part of GNAT Pro documentation.

GNATcoverage RM

AdaCore: GNATcoverage Reference Manual.

CHAPTER

TWO

TOOL QUALIFICATION PLAN

2.1 Compliance to guidance

This section contains compliance matrix with the guidance contained in section 12.2 of DO-178B.

Note: in the following, we always use the term "verification tool" to indicate the software level of the tool following the DO-178B terminology. In this context, this term is equivalent to TQL5 for DO-178C.

Table 2.1: Section 12.2

Section	Data	Notes
12.2a	verification tool	see Tool type
12.2b	not applicable	verification tool
12.2c	see *Software Configuration Management Plan* and see *Software Quality Assurance Plan*	
12.2.1	not applicable	verification tool
12.2.2	see "Tool Operational Requirement and Test cases document", see "Tests Results document", and GNATcoverage Qualified Interface	
12.2.3a	to be provided by the applicant	See *User Activities*
12.2.3b	CC2 see *Software Configuration Management Plan*	verification tool
12.2.3c	not applicable	verification tool
12.2.3.1	not applicable	nonetheless we provide *Tool Qualification Plan* (this document)
12.2.3.2a	see "Tool Operational Requirement and Test case document"	
12.2.3.2b	see "GNAT Pro User's Guide" and "GNATcoverage User's Guide" and "GNATcoverage README"	
12.2.3.2c	To be provided by the user	See *User Activities*
12.2.3.2d	not applicable	verification tool
12.2.4	to be provided by the applicant	see *User Activities*

2.2 Tool Overview

GNATcoverage is a tool designed to assess the source level structural coverage achieved by a testing campaign on software. It operates without any instrumentation of the original program, on code generated by the same compiler as the one producing the executable deployed on the target hardware.

Detailed information on the tool capabilities and characteristics is available in the GNATcoverage User's Guide as well as in [AE09] an [ERTS10] papers.

2.3 Sought Certification Credit

GNATcoverage aims at automating the structural coverage assessment activities required by the Software Verification Process of DO-178B and described in section 6.4.4.2 and table A7, objectives 5, 6 and 7 depending on the software level of the embedded application. Depending on the parameters passed to the GNATcoverage qualified interface, see GNATcoverage Qualified Interface, a single execution of GNATcoverage may satisfy one, two or all of the objectives above.

2.4 Tool type

Given the sought certification credit, GNATcoverage is qualified as a **verification tool**.

2.5 GNATcoverage qualified interface

To obtain reports suitable for use as certification evidence, applicants shall use GNATcoverage as follows:

- Build the object code composing the application under test with the GNAT Pro toolchain identified in the Tool Operational Requirements obeying to the compilation options and coding standard rules documented in the Operational Environment section.
- Build the test code and test harness and link it with the relevant application objects if necessary to form one or several test executables. The test code does not need to be compiled with the switches described in the Operational Environment section, nor does it need to comply to the coding standard.
- Obtain as many execution trace files (<TRACE>) as needed by running the test executables (<APP>) within the instrumented execution environment:

```
xcov run --target=<TARGET> --level=<LVL> <APP> -o <TRACE>
...
xcov run --target=<TARGET> --level=<LVL> <APP> -o <TRACE>
```

- Produce a single <REPORT> file (format documented in the GNATcoverage User's Guide), consolidating the various coverage information associated with the various execution traces of interest:

```
xcov coverage --annotate=report --level=<LVL> --scos=@<alis.list> @<traces.list> -o <REPORT>
```

where:

- <APP> is test executable
- <LVL> designates the coverage criteria to assess. The value used for <LVL> depends on the software level on the embedded application:
  - for level C applications, <LVL> shall be equal to "stmt". In this case, a single execution of GNATcoverage produces statement coverage data
  - for level B applications, <LVL> shall be equal to "stmt+decision". In this case a single execution of GNATcoverage produces statement coverage data *and* decision coverage data.
  - for level A applications, <LVL> shall be equal to "stmt+mcdc". In this case, a single execution of GNATcoverage produces statement coverage data *and* decision coverage data *and* mcdc coverage data.
- <REPORT> is the output file containing the GNATcoverage report
- <TARGET> identifies the target platform (as in the GNAT Pro toolchain prefixes, e.g. powerpc-elf);
- <TRACE> is the output file containing the execution trace
- <alis.list> is a text file containing the list of GNAT Pro ALI file names associated with the units for which coverage is assessed

- <traces.list> is a text file containing the list of execution traces to operate on.

2.6 Environment Equivalence

Qualification data is produced both in the environment where qualification activities are performed and in the environment where the qualified tool is used (see see *User Activities*). For the whole set of qualification material to be consistent, those 2 environments must be equivalent. The equivalence of the following items is deemed sufficient to establish equivalence of environments:

1. the GNAT Pro executable name, version number and host operating system;
2. the list of GNAT Pro compilation switches;
3. The GNATemulator executable name, version number and host operating system;
4. the GNATcoverage executable name, version number and host operating system.

If all items above are the same in the 2 environments, then they are considered equivalent for the purpose of GNATcoverage usage.

2.7 User Activities

In order to finalize the qualification of GNATcoverage, the final user needs to perform a set of activities, either in the scope of GNATcoverage qualification or GNATcoverage usage.

GNATcoverage Qualification

1. **Reference GNATcoverage in the PSAC.** The user needs to:
   - identify GNATcoverage as a verification tool that needs to be qualified;
   - identify the compilation options for GNAT Pro.
2. **Delivery Acceptance.** On delivery of GNATcoverage and its qualification kit, the user shall assess the representativeness of the verification environment with the operational environment.
3. **Description of the tool operational environment**, see section 12.2.3.2c of [DO-178B]
4. **To establish Archive, retrieval, and release** (objective 7.2.7), see see the compliance matrix for table A-8 in *Software Configuration Management Plan*.
5. **Provide a tool qualification agreement**, see section 12.2.4 of DO-178B.

GNATcoverage usage

1. **Tool Installation in Operational Environment.** The user needs to install the tool in the Operational Environment.
2. **Check correct usage of GNATcoverage .** For GNATcoverage results to be used in a certification context, the used tool interface must comply with the GNATcoverage qualified interface.
3. **Update Environment Configuration Index.** The delivery file shall be included in the Environment Configuration Index; the Tests Results shall be included as well.
4. **Update the Software Accomplishment Summary (SAS).** The SAS need to be updated:
   - For objective of table A7, objective 5, 6, or 7 depending on the criticality level of the embedded application.
   - For qualification status of GNATcoverage.

This page is intentionally left blank.

CHAPTER

THREE

SOFTWARE CONFIGURATION MANAGEMENT PLAN

GNATcoverage is qualified as a verification tool. As such such most configuration management activities are not mandatory, in particular:

Table 3.1: Compliance matrix for Table A-8

Item	Description	Ref.	Notes
1	Configuration items are identified.	7.2.1	See List of configuration items.
2	Baselines and traceability are established.	7.2.2	Not required for verification tools
3	Problem reporting, change control, change review, and configuration status accounting are established.	7.2.3, 7.2.4, 7.2.5, 7.2.6	Not required for verification tools
4	Archive, retrieval, and release are established.	7.2.7	User activity, see *User Activities*
5	Software load control is established	7.2.8	Not required for verification tools
6	Software life cycle environment control is established	7.2.9	Not required for verification tools

3.1 List of configuration items

- Tool operational requirements
- Test cases
- Test data
- Test execution results

3.2 AdaCore internal configuration management process

This section contains additional information of AdaCore internal configuration management process.

3.2.1 Configuration Management Methods

Configuration Management is technically implemented via a Subversion repository. The life cycle of each artifact is automatically tracked in the repository. Informal email-based discussions about a precise artifact are also tracked. Each major artifact (requirement, qualification reports, etc.) is associated to a unique ID which is referenced in each email discussing the evolution of that precise artifact. In this manner, it is fairly easy to

reconstruct the whole evolution of an artifact a posteriori simply by looking at SVN commits and email referencing its ID in their subjects. The mechanism used to implement this tracking uses the Customer Management System deployed at AdaCore: such technology has been widely used for the last ten years.

Official baseline production

Official baselines are generated on a customer-specific delivery for a precise operational environment. A specific folder and .zip file is created for each official release.

Archiving

All repositories and mail servers are redounded with machines physically located in Paris (France) and New York (The United States). This increases our confidence on the durability of qualification data.

3.2.2 Activities

Artifact identification

Each atomic artifact (single requirement, test case, expected output, compilation unit) is located in a single physical file, so as to permit its atomic tracking.

Plans and documentation

All documentation material is under Configuration Management Control. Each single part (each section) of each document is tracked automatically.

Development and Verification Artifacts

The tracked development and verification artifacts are the configuration item at List of configuration items plus:

- Build/test infrastructure
- QA reports

Quality Assurance Reports

Quality Assurance Reports are atomically tracked exactly like any other textual artifact of GNATcoverage qualification material. Quality assurance reports are specific for each tool released and their lifecycle is tracked on a release-specific basis.

Open problems identification

Open problems are tracked via emails. Each email is associated to a unique problem identified by a unique ID. Each problem is assigned to a single entity of the Development or Qualification team. The unique ID identify the open problem within a database which permits to track its evolution and status (open/closed). All emails are saved in a database and it is possible to query it to retrieve all mails related to a precise open problem.

CHAPTER

FOUR

SOFTWARE QUALITY ASSURANCE PLAN

We remind here that GNATcoverage is qualified as a verification tool. As such such most configuration management activities are not mandatory, in particular:

Table 4.1: Compliance matrix for Table A-9 of DO-178B

Item	Description	Ref.	Notes	Activity
1	Assurance is obtained that software development and integral processes comply with approved software plans and standards.	8.1a	For verification tools, this is limited to the compliance of tool processes with approved plans	Reading of GNATcoverage Qualification Material: Plans, Inspection of qualification data (by sampling)
2	Assurance is obtained that transition criteria for the software life cycle processes are satisfied.	8.1b	Not required for verification tools	n/a
3	Software conformity review is conducted.	8.1c, 8.3e, 8.3f	Items 8.3a, 8.3b. 8.3c, 8.3d, 8.3g, 8.3h and 8.3i are not required for verification tools or CC2; 8.3f is also not required because verification tool qualification is supposed to be black-box	Tool Conformity Review

4.1 Quality Assurance Activities

This sections contains a description of the Quality Assurance activities in terms of objective and output.

4.1.1 Reading of GNATcoverage Qualification Material: Plans

The quality assurance on plans is not specific to a precise client: reports are thus simply identified by date.

- **objectives:** to assess the compliance with qualification objectives
- **output:** QA Reading report (qa/DDMMYYYY/qa_plans.doc)

4.1.2 Inspection of qualification data (by sampling)

Inspection of Tool Operational Requirements (by sampling)

The quality assurance on tool operational requirements is specific to a precise client and operational environment: reports are thus identified by the client name and a date.

- **objectives:**
  - check the accuracy, completeness and consistency with plans.
- **output:** QA inspection report (qa/<CLIENT>/DDMMYYYY/qa_tor.doc).

Inspection of Test Cases (by sampling)

The quality assurance on tool operational requirements is specific to a precise client and operational environment: reports are thus identified by the client name and a date.

- **objectives:**
  - check the accuracy of test cases, in particular the representativeness of target Ada constructs.
- **output:** QA inspection report (qa/<CLIENT>/DDMMYYYY/qa_tor.doc, the same file used for the review of Tool Operational Requirements).

Inspection of test execution results

The quality assurance on tool operational requirements is specific to a precise client and operational environment: reports are thus identified by the client name and a date.

- **objectives:**
  - Check the results of test execution
  - In the case tests failed, it is necessary to investigate whether the source of error is:
    * A misbehaviour of the infrastructure used to run tests and compare actual results to expected results: in this case, the GNATcoverage Qualification Team is in charge of reporting and fixing the problem.
    * A bug in the GNATcoverage implementation: in this case, the GNATcoverage Development Team is in charge of reporting and fixing the problem.
    * A reasonable limitation of the tool: in this case, the GNATcoverage Qualification Team is in charge of reporting and justifying the problem as part of the know limitations of the tool.
- **output:** QA inspection report (qa/<CLIENT>/DDMMYYYY/ qa_test_execution.doc)

4.1.3 Tool conformity review

The quality assurance on tool operational requirements is specific to a precise client and operational environment: reports are thus identified by the client name and a date. The conformity review takes in input a packaged and qualifiable release of GNATcoverage.

- **objectives:**
  - Record and approve software requirements deviations (8.3e).
- **output:** QA inspection report qa/<CLIENT>/DDMMYYYY/ qa_conformity.doc)