

IPL Testing Tools and Defence Standard 00-55

Executive Summary

This paper shows how AdaTEST and Cantata++ can be used to assist with the development of software to Defence Standard 00-55. In particular, it shows how AdaTEST and Cantata++ can be used to meet the verification and testing requirements of the standard. It is also shown that AdaTEST and Cantata++ have been produced to a standard of sufficiently high integrity that their use will not compromise the safety integrity of the software being tested.

The material presented here is suitable for inclusion in a justification for the use of the products on a safety-critical software development.

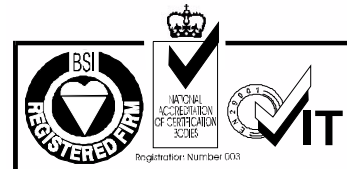
IPL is an independent software house founded in 1979 and based in Bath. IPL's QMS is accredited to ISO 9001 and TickIT (ISO 9000-3). Both AdaTEST and Cantata++ have been produced to these standards.

Copyright

This document is the copyright of IPL Information Processing Ltd. It may not be copied or distributed in any form, in whole or in part, without the prior written consent of IPL.

*IPL
Eveleigh House
Grove Street
Bath
BA1 5LR
UK*

*Phone: +44 (0) 1225 475000
Fax: +44 (0) 1225 444400
Email: tools@iplbath.com*



Certificate Number FM1589

*Last Update: 30/05/2002 14:20
File: 0055.doc*

1. Introduction

This paper details how the IPL Software Testing Tools, AdaTEST and Cantata++ can be used to support a software development which uses the Defence Standard 00-55, Requirements for Safety-Related Software in Defence Equipment. Section 2 of this paper provides a generalised overview of the functionality of the tools. Section 3 of the paper matches the tools' capabilities against the requirements of the standard. It will be seen that AdaTEST and Cantata++ are not intended to support the formal analysis and design validation parts of the standard, so additional detail is given to those sections which relate to software testing. Finally, section 4 gives further supporting evidence on the tools' internal integrity for safety-related software developments.

2. General Description of AdaTEST and Cantata++

AdaTEST and Cantata++ are the generic names for the families of products developed by IPL for the testing of Ada, C and C++ software. There are four products currently available:

AdaTEST, for testing Ada 83

AdaTEST 95, for testing Ada 95 (and Ada 83)

Cantata, for testing C (and simple C++)

Cantata++, for testing C++

These tools have been expressly created to assist in the Software Unit and Integration testing phases of development projects using these languages. The aim is to assist developers in producing tests which are automated, repeatable, and can be run on any host or target platform. The precise functionality and availability of the tools will vary over time, so please refer to IPL for the current detailed status of the products.

The functionality of the tools can be summarised in the following table:

Dynamic Testing capability

- The production of test drivers for executing the software under test through a series of planned test cases.
- The setting of automated 'checks' on the test case outputs, to verify that these match the expected results.
- The creation/generation of simulations for external subprograms or objects, to better enable the software under test to be verified in isolation from other components in the system.
- The detection of expected and unexpected exception raising.
- Timing Analysis, to verify that software execution times match performance requirements.
- The generation of a test result summary which states whether the overall test passed or failed.
- The generation of full test results output which includes diagnostic information on failed checks.

Test Coverage Analysis

- Determination of the effectiveness of dynamic testing by measuring the coverage of key elements in the software under test. These elements can include any or all of the following (product dependent, refer to IPL for details): entry points, statements, decision and branches, conditions, exceptions, data values.
- The potential to cause overall test failure if any coverage measurement does not reach a preset minimum level.
- The generation of coverage statistic reports, which provide execution profiles including the identification of unexecuted code elements.
- The generation of trace reports to assist debugging.

Static Analysis

- The generation of source code metrics relating to language construct usage and software complexity.
- Use of metrics for pass/fail checks during unit and regression testing activities.
- The availability of these metrics in comma-separated value (csv) format, for conversion into graphical or database forms using suitable tools.

3. Matching Tool Capabilities Against the Requirements

This section provides a systematic analysis of the requirements of Def-Stan 00-55 section by section, detailing those parts where it is felt that AdaTEST and Cantata++ can make a positive contribution. It is organised to correspond to the sections of the standard, apart from section 3.5 which is exclusively given over to the details of Requirement 37 (Testing and Integration). The references given in *(parentheses)* relate directly to the requirement numbers in the standard.

Section 2. Safety Management

Section 2 of 00-55 is primarily concerned with the project defining how it intends to meet the standard's requirements. As such AdaTEST and Cantata++ do not directly have much to offer. The tools can however be referenced in minor ways:

- The Software Safety Case can include reference to test results data generated by AdaTEST and Cantata++ (7.2.7).
- The Software Safety Arguments can include reference to testing carried out by AdaTEST and Cantata++ (7.3.2 b).
- The Software Safety Records Log will be where the test results data can be contained or referenced (9).

Section 3. Roles and Responsibilities

This section's key requirement for a V&V Team independent of the Design Team (17). This team carries out or reviews all testing (17.4). AdaTEST and Cantata++ have a useful role to play here because their use leads to the production of repeatable tests, which generate full test result reports. Hence, the task of unit testing can be carried out either by the V&V team themselves, or by the Design/Code team and then repeated and reviewed by the V&V team.

Section 4. Planning Process

There are various ways in which AdaTEST and Cantata++ are relevant to the Planning Process section:

- Their use can be identified as one of the software tools on the project (21.5 d).
- They can be referenced during planning of the activities for the V&V team (24.2 b), and the test results from AdaTEST and Cantata++ can be used to define the acceptance criteria for SRS items (24.4).
- Their planned use should be mentioned in Configuration Management planning (25.3 c).
- The facilities of AdaTEST and Cantata++ can be referenced directly when describing how the following methods will be implemented on the project: static/subset analysis (26.2 c) and dynamic testing (26.2 d).
- When choosing which high level language to be used (28) it is to be noted that AdaTEST and Cantata++ are both intended for use with high-level languages, and furthermore that they contain static analysis facilities which can be useful in subset analysis (28.1). If AdaTEST and Cantata++ are to be used for this purpose they can be referenced as such in the Code of Design Practice (28.4).
- Under Selection of Tools (29) it may be useful to reference the extensive previous use of AdaTEST and Cantata++ in Safety-Critical applications, to justifying their use (29.1 and 29.2). This document is intended to assist in matching of the tools' capabilities against the requirements of the standard (29.5.1). AdaTEST and Cantata++ have much supporting evidence to offer in the areas of usability, interoperability, stability, availability, support and familiarity (29.5.2).
- The existence of unreachable code (30.2) can be detected with both AdaTEST and Cantata++ dynamically and, in some cases, statically.
- Both AdaTEST and Cantata++ have useful facilities to offer in the area of reverse engineering of previously developed software which was not developed to Def-Stan 00-55 (30.4). Specifically, these can include static analysis of large volumes of source code, and coverage analysis of unit tests not originally carried out with either AdaTEST or Cantata++.

Section 5. SRS Development Process

The use of AdaTEST and Cantata++ results can be referenced when recording how compiled code was tested (32.2.4), and also when detailing traceability (32.3.2) between detailed design requirements and unit tests.

Evidence from unit testing can be used to justify object code correctness (36.6.3). Test evidence can be used to support the case for the compiler (36.6.3 a) and for test coverage at the source level (36.6.3 c)

Testing and Integration (37)

AdaTEST and Cantata++ can be used for testing all properties and functions (37.1.1 a), on the target hardware as well as host (37.1.1 b). They can demonstrate that dynamic and performance requirements are met (37.1.1 c), and also be used to generate 'stress conditions' (37.1.1 d).

Coverage metrics and test design can be expressed in ways suitable for AdaTEST and Cantata++ (37.1.2 c).

AdaTEST and Cantata++ are well suited to a test design approach based on the specification of test case input and expected outputs (37.1.3). Indeed AdaTEST and Cantata++ test scripts can be used as self-documenting Test Specifications.

Test scripts for AdaTEST and Cantata++ are easily reviewed (37.1.5).

Configuration control is easy to apply to AdaTEST and Cantata++ scripts (37.1.7) which are simple ASCII files.

While testing with AdaTEST and Cantata++ can be carried out on a host machine and can involve the use of instrumented code for coverage analysis, this is not compulsory and there remains the option to run the test on the target with non-instrumented code (37.1.8 a, b).

Target tests with AdaTEST and Cantata++ can be run under an emulator (37.1.9).

AdaTEST and Cantata++ are designed to operate identically in host and target environments and facilitate the production of completely portable tests between these environments (37.1.10).

Test output from AdaTEST and Cantata++ is in the form of an ASCII file and is very suitable for inclusion in the Test Record (37.1.11).

Check diagnostics in AdaTEST and Cantata++ test results always give a statement of the actual and expected values (37.1.12) though it is left to the Design Team to explain any discrepancies.

AdaTEST and Cantata++ tests are highly repeatable, making the revision process easy (37.1.13).

AdaTEST and Cantata++ are suitable for use at both unit and integration test levels (37.3.1).

AdaTEST and Cantata++ can be used to measure all forms of test code coverage (37.3.2 a, b, c, e, f).

The products support integration coverage measures which assist in measuring module interface coverage during integration testing (37.3.3). All products supports entry

point coverage which is the means of ensuring that all of a given set of subprograms have been executed at least once.

AdaTEST and Cantata++ can be used (depending on the circumstances) for various aspects of System Testing (37.3.4-6).

Section 6. Certification and In-Service Use (& B1, Documentation)

The test results generated by AdaTEST and Cantata++ can provide evidence for the Safety Case Log and the Software Safety Records Log (38.3).

The test scripts used by AdaTEST and Cantata++ are reusable, making them suitable for use in a code modification and retest situation (42.7 d, g).

The formats used by AdaTEST and Cantata++ for both input and output make it easy to use them in the generation of Test Records, Test Schedules and Test Reports (*B1 r, s, t*).

4. Tool Integrity and Development Standards

A number of arguments can be offered to justify the use of AdaTEST and Cantata++ in a safety-related software project. The first is that all IPL tools have been developed according to the IPL Quality Management System (QMS), which has been accredited to ISO 9001 and ISO 9000-3.

The core of the AdaTEST product was itself developed as a safety-critical product employing such recognised techniques as hazard analysis, and independent audit, use of a 'safe' language subset, in addition to all the normal requirements of the IPL QMS. Details of the methods used can be found in the AdaTEST project documents.

Since their release the AdaTEST and Cantata++ products have successfully been qualified through a number of customer audits, principally for use on RTCA/DO-178B projects at Level A (Safety-Critical). The customers involved include Boeing Aircraft, GEC Marconi, Rolls Royce AeroEngines, Sextant Avionique and Ultra Electronics. Reports on these audits can be inspected by arrangement with IPL. Customers are encouraged to conduct their own audit of the tools to justify their use on safety-related projects.

5. Conclusion

AdaTEST and Cantata++ support all the dynamic testing requirements of Defence Standard 00-55, in a simple to use and well-integrated manner. They facilitate a high degree of automation of dynamic testing and other verification techniques required for effective use of the standard.

AdaTEST and Cantata++ have been developed to the highest practical standard for software verification tools. They have both proven integrity and reliability through extensive use.

It is believed that AdaTEST and Cantata++ are the only tools to offer this comprehensive functionality and the only testing tools developed to such high standards. However, this does not preclude the use of AdaTEST and Cantata++ for testing software which is not for safety critical use.