

# IPL Testing Tools and ECSS E-40 Space Software Engineering

## Executive Summary

This paper describes how Cantata and AdaTEST can be used to assist with the verification and validation of software according to the ECSS E-40 Standard on Space Software Engineering. In particular it shows how the tools can be used to support the testing aspects of these guidelines. It also shows that the tools have been produced to a high quality standard such that their use for dynamic testing will not compromise the safety and integrity of the software being tested.

IPL is an independent software house founded in 1979 and based in Bath. The company has carried out work in the area of safety-related systems since 1987, and its Software Code of Practice has been approved by several major customers. IPL achieved accreditation to the ISO 9001 quality standard in 1988 and ISO 9000-3 in 1992. Both Cantata and AdaTEST 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 [ipl@ipl.com](mailto:ipl@ipl.com)*



*Certificate Number FM 01580*

## 1. Introduction

This paper details how the IPL software testing tools, Cantata and AdaTEST can be used to support a software development process which is aiming to follow the ECSS E-40 Standard on Space Software Engineering. The document issue used is dated November 2003. This is referred to as ‘the standard’ in the rest of this paper.

Part 2 of the paper provides a generalised overview of the functionality of the tools. Part 3 of the paper matches the tools’ capabilities against the requirements of the standard. It will be seen that Cantata and AdaTEST are mainly useful in supporting the activities described in the ‘Software Design and Implementation’ (5.5) and ‘Software Verification’ (5.8) sections of the standard.

Finally, Part 4 gives further supporting evidence for the case on the tools’ internal integrity for safety-related software developments in general.

## 2. General Description of Cantata and AdaTEST

Cantata and AdaTEST are the generic names for the families of products developed by IPL for the testing of C/C++ and Ada software. In fact, at this point in time (November 2005) there are two products available:

Cantata++, for testing C and C++

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

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 the developers to produce 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 generic functionality of the tools can be summarised as follows:

### **Dynamic Testing capability.**

- The production of test drivers for executing the software under test through a series of planned test cases. The test drivers are generated automatically from test data supplied by the user in the native language.
- The setting of automated ‘checks’ on the test case outputs, to verify that these matched the expected results. These are available for both standard and user-defined types.
- 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. Two techniques are available: ‘stubbing’, for all IPL tools, and ‘wrapping’, only for Cantata++
- 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.**

- The measurement of the effectiveness of the test cases by the computation of their coverage of elements of the software under test. These elements can include any or all of the following (slightly product dependent, refer to IPL information for details): entry points, statements, decision/branches, conditions, exceptions, data values ('always' and 'at least once').
- The potential to cause an overall test failure if any coverage achieved 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.
- All of the above but with tests driven by means external to the tools. IPL calls this 'standalone' coverage analysis.

### **Static Analysis.**

- The generation of source code metrics relating to the following: language construct usage, software complexity using industry-agreed definitions of the term, file and other related metrics.
- The placement of these metrics into an ASCII or CSV file suitable for code review in the case of AdaTEST, or CSV only for Cantata++. The data in CSV format can be used by any suitable external tool such as a spreadsheet for a variety of downstream analyses and presentations.
- The availability of these metrics for pass/fail checks during unit and regression testing activities (not Cantata++).

## **3. Matching Tool Capabilities Against the Standard's Requirements**

This part is a systematic analysis of the recommendations of the guidelines section by section, and the detailing of those parts where it is felt that Cantata and AdaTEST can make a positive contribution. It is organised according to the sections of the standard. The references given in italics relate directly to section numbers, names and also text quoted from the guidelines.

### ***Section 5.5. Software Design and Implementation Engineering Process***

#### ***5.5.2 Design of software items***

Designs for software units are created such that these units can be 'coded, compiled, and tested'. IPL advises that attention be paid to designing the software in such a way

that code produced is capable of being tested in accordance with the standard. Please check with IPL if advice is needed.

#### *5.5.2.2 Software Interfaces Detailed Design*

From a testing point of view it is important that the interfaces to all software units are properly understood and specified. Only in this case will it be possible to provide the external interface simulations that are usually needed for successful testing.

#### *5.5.2.9 and 5.5.2.10 Software unit and integration test plans*

In any level of software test planning it will be expected to see mention of any tools which will be used to assist the process. Cantata and AdaTEST have a long history of use of this purpose in ESA and other space projects.

### **5.5.3 Coding and Testing**

#### *5.5.3.1 Software unit test data*

This section of the standard identifies the need to produce test procedures and data for testing each software item. At the very least IPL recommend that this should include a note on how the external interfaces are to be treated (real code, stub, or other simulations), and outline a series of test cases each intended to verify some specified aspect of the unit's intended functionality. Test data should ideally include values for input test data to be used (e.g. parameter and global data values). Boundary cases are particularly recommended once the unit's core functionality has been demonstrated. Expected output values are also helpful for software testers to have provided.

#### *5.5.3.2 Software unit testing*

This simply states that units shall be tested to ensure that they satisfy their requirements, and that test reports shall be produced.

This is precisely the role for which Cantata and AdaTEST have been produced, and any user should have no problem in using the tools for this purpose. The test result files generated when Cantata or AdaTEST tests are run is ideal for submission as either part or the whole of the test reports.

### **5.5.4 Integration**

#### *5.5.4.1 and 5.5.4.2 Software integration testing*

In much the same way as software unit tests should be planned in advance, the same is true of software integration tests. Cantata and AdaTEST can offer significant assistance in doing the tests and may usefully be referred to in the planning process.

## **Section 5.6 Software Validation Process**

This section contains many references to testing. It is IPL's opinion that testing is a verification activity so these references are possibly misplaced. Please see the points in Section 5.8.

## **Section 5.8 Software Verification Process**

#### 5.8.2.1 Determination of the verification effort

Compared with manual methods of verification IPL tools offer considerable productivity advantages. Any project wishing to minimise verification effort (and hence cost) should consider use of these tools. The products have a reputation based on performance for being able to support development to the highest levels of criticality, including safety-critical.

#### 5.8.2.2 Verification process, methods and tools

As previously mentioned the software test planning stage will normally be expected to name any tools used. At this stage this could be amplified to detail which tools are to be for which verification activities, and in which way. Cantata and AdaTEST can be used flexibly for different purposes and in different ways. Please consult IPL for suggestions on these points.

#### 5.8.2.3 Selection of organisation responsible for verification

If it is intended to use any outside or independent software verification resource, IPL will be pleased to consider this activity, using IPL tools or others if preferred. The company has a track record in performing this kind of work both in the Space industry and outside.

#### 5.8.2.4 Verification plan

A master plan for all verification activities is to be produced. It is recommended that this is the ideal point for consideration of what constitutes a 'unit' for the purposes of unit testing and what levels of software integration testing are to be done should be considered. The use of tools such as Cantata or AdaTEST can be put forward at this point and the precise purpose for and manner in which they shall be used is clarified.

#### 5.8.3.4 Verification of code

Cantata and AdaTEST can be useful in a number of ways:

- *The code is ... testable, correct, and in conformity to software requirements and coding standards.*
  - If code is not testable this will quickly be discovered by Cantata or AdaTEST. Re-design may be required. Correctness and conformity to requirements can be demonstrated to a high degree through testing with Cantata and AdaTEST. Conformance to coding standards can be at least partly checked by use of code metrics produced by Cantata and AdaTEST.
- *The code implements proper event sequence, consistent interfaces, correct data and control flow, completeness, appropriate allocation timing and sizing budgets, and error handling.*
  - The majority of these points can be demonstrated through appropriately designed test cases.

- *The code implements safety, security, and other critical requirements correctly.*
  - As above, these aspects can in the main be demonstrated through appropriately designed test cases.
- *External consistency with the requirements ... of the software item.*
  - This seems to be mainly a need to test items in a ‘black-box’ manner. Cantata and AdaTEST can be used to test software in both black-box and white-box manners.
- *Internal consistency between software items.*
  - This seems to refer to a need to check software interfaces, which is a common aspect of software unit and integration testing.
- *Absence of run-time errors.*
  - It is difficult to demonstrate this by testing alone. Use of static analysis tools, such as Klocwork may help. Refer to IPL if further information is required.
- *Test coverage of units.*
  - Measurement of test coverage is an integral part of Cantata and AdaTEST functionality. Many different types of coverage are supported by the tools. These include entry-points, statements, decisions, and conditions. Coverage output from the tools can include test pass/fail checks on whether a pre-set level of coverage was achieved, and coverage statistics which show which parts of the code were executed and how many times.

#### *5.8.3.5 Verification of Software Integration*

Depending on the precise nature and circumstances of the proposed integration testing then Cantata and AdaTEST may be useful with at least some of the activities in this section. Typically the tools can be used for ‘cluster’ testing (groups of software modules), task testing, and to a lesser extent sub-system testing.

#### *5.8.3.7 Verification of test specifications*

When providing traceability information between software requirements and tests it is useful to be able to reference specific test cases within specific Cantata and AdaTEST scripts.

### **Section 5.10. Software Maintenance Process**

#### *5.10.4.3 Invoking of software engineering processes for modification implementation*

This section describes the need to test both modified (for correct changes) and unmodified (for unwanted changes) parts. The use of Cantata or AdaTEST will help in both aspects. Batch mode execution of sets of tests are easy to create and run, thus facilitating in particular that unwanted changes have not occurred when modifying other parts of the software.

#### *5.10.6.2 Migration planning and execution*

When moving code from one platform to another it is important that the units be re-tested in the new environment. Use of Cantata and AdaTEST can be very beneficial in this case because the tests are extremely portable, and can usually be moved across platforms with no modification at all.

#### **4. Tool Integrity and Development Standards**

A number of arguments can be offered to justify the use of Cantata and AdaTEST in a safety- or mission-critical 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.

Since their release the Cantata and AdaTEST products have successfully been through a number of customer audits, principally for use on safety-related avionics projects. Reports on these audits can be inspected by arrangement with IPL. The products have also been used on a large number of space projects, principally but not only, ESA projects. Customers are actively encouraged to conduct their own audit of the tools.