



Cantata++

intelligent testing

**An Overview for
DO-178B
Tool Qualification**



Contents

Overview of IPL and *Cantata++*

Tool Qualification Case

- Qualification Requirements and Credentials
- Tool Qualification Pack
- IPL Quality Involvement
- Cantata++ Tool Development
- Cantata++ Technical Support
- Further Information



Trusted Consulting & Solutions Company

- 30 year track record
- 260 staff, €30m+ turnover
- HQ in Bath, UK
- High-stakes, business and mission critical
- Real Time Systems:
 - Aerospace & Defence
 - Police
 - Telecoms
 - Automotive & Rail Transport
 - Medical

Testing Tools:

- 20 year track record
- Grew out of services business



Cantata++
intelligent testing



BUILT ON
eclipse™

Cantata++

intelligent testing

What is *Cantata++*?

Cantata++ is a software verification tool for host and target environments providing:

- **Dynamic Testing**

 - Unit and Integration test execution

 - Automated results checking

 - Test documentation / evidence of testing

 - Test repeatability

- **Structural Coverage Analysis for Software Levels C to A**

 - Entry-Point, Statement, Decision, Boolean Operator, Boolean Operand Effectiveness (Masking and Unique Cause)

- **Static Analysis Metrics**

- **Ancillary Tools**

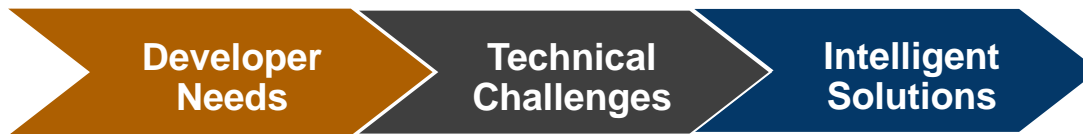
 - Built on Eclipse user interface for test scripting

 - Advanced test and coverage diagnostics

Unit/Integration Test – 6 Tool Challenges



- Confidence that my tests on the target cover the important code
- Having a full tool-kit to efficiently test the way I need to
- Making it simple for me to test object interactions with the rest of the system
- Finding time to create reusable test harnesses to test all my functionality
- Not fighting the tool, but working seamlessly with what I already have
- Knowing where to focus my test effort and how much work it will be

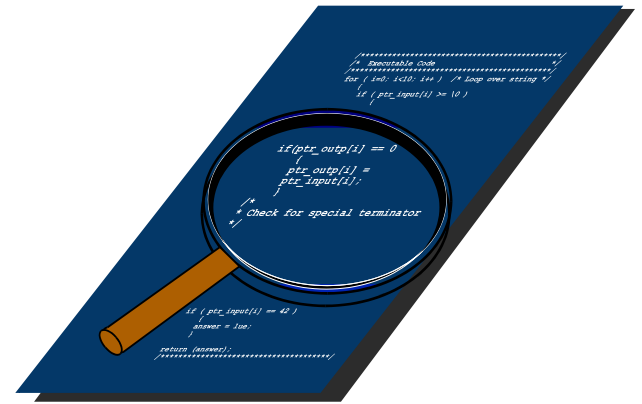


Cantata++

1 Code Complexity

Challenges:

- Analyze code before testing
- Identify maintainability
- Aid test planning



➔ **Cantata++** measures...

● Procedural and OO Metric Sets

Myers

McCabe

Halstead

Hansen

McCabe Object Oriented

MOOSE

MOOD

QMOOD

Robert Martin

Bansiya's Class Entropy

● Reports in .csv format

2 Tool Reliability

Challenges:

- Reliability and Maintainability
- Market use
- Documentation, support, training etc
- Future Proofing
- Integration with environment



IPL Quality Management System
LRQA Certificate 4006360



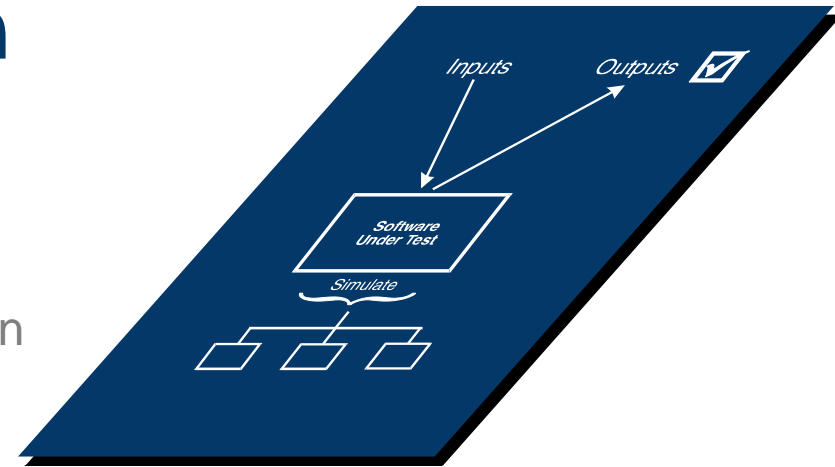
➔ **Cantata++** reassures with...

- Developed under IPL certificated QMS
 - DO-178B tool qualification package
 - Cantata++ multi-target deployment technology
- Widespread market use
- Comprehensive documentation
- Direct access to real engineers for technical support
- On-site product training and consultancy
- Product road-map

3 Test Harness Creation

Challenges:

- Test harness driver for individual units
- Structured auto-repeatable test cases
- Automated checking and results production



➔ **Cantata++** automates through ...

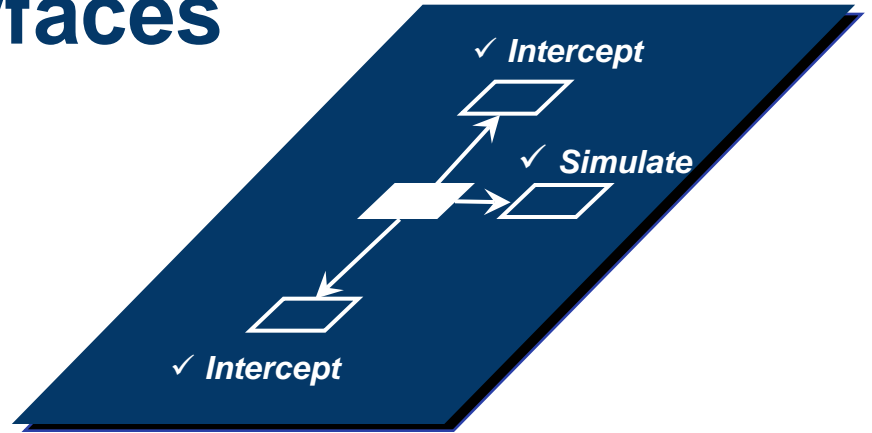


- Parse of project code
- Automated structured test script
 - ☑ case per function/ method
 - ☑ parameters, call order, global data, returns
 - ☑ test case independence
- Full featured check library
- Positive and negative checking of global data
- Automated results production

4 Control of Call Interfaces

Challenges:

- Isolation of units from system
- Order of calls
- Different behavior on calls
- Checking/modifying parameters

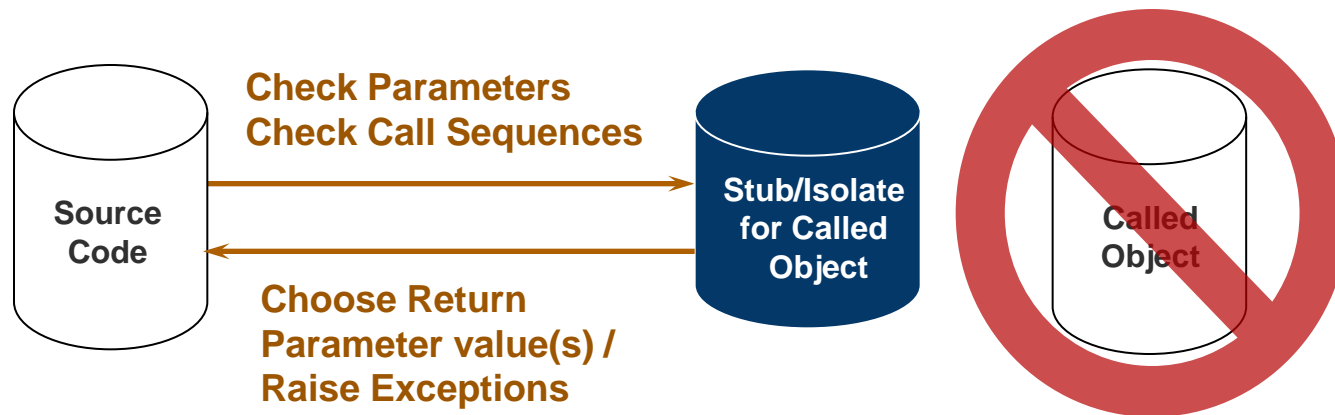


➔ ***Cantata++*** enables control by...

- Knowledge of calls made
- Automated generation of
 - Stubs and Isolates to simulate calls
 - Wrappers to intercept calls
- Programmable instances for each call
- Flexible call sequence control
- Automated checks on parameters

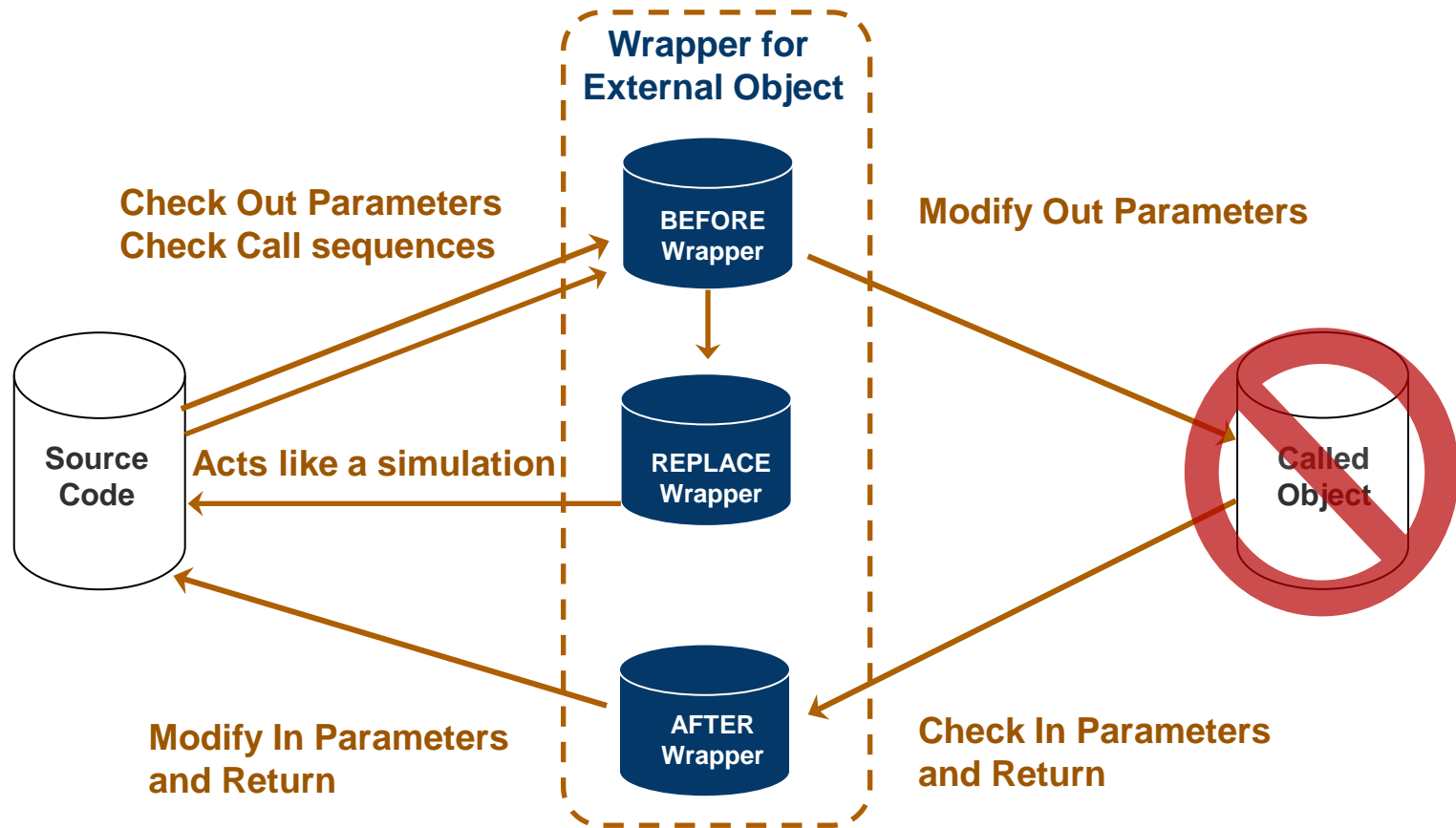
***Cantata++* Stubs and Isolates**

- A function/method inside test script with programmable instances
- Stub replaces called object at link time
- Isolate replaces specific calls to called objects at link time



Cantata++ Wrapping

- A function/method inside test script with programmable instances to select Before-After or Before-Replace Pairing
- Intercepts specific calls to called objects at link time



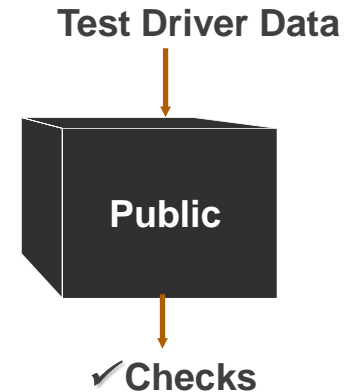
Call Interface Control Flexibility

Action	Stub	Isolate	Wrap
Called object simulated	Must	Must	✓
Called object included in link	-	✓	Must
Check inputs to called object	✓	✓	✓
Check outputs for called object	-	-	✓
Check call order	✓	✓	✓
Check or Set any data when object called	✓	✓	✓
Check or Set any data when object returns	-	-	✓
Set outputs from called object	Must	Must	✓
Modify inputs to called object	-	-	✓
Modify outputs from called object	-	-	✓
Control system calls	-	✓	✓
Control variadic function calls	✓	✓	-
Control class method calls	✓	-	✓

5 Flexibility: Black Box Testing

Challenges:

- Large data input sets
- Checking large output sets
- Robustness tests



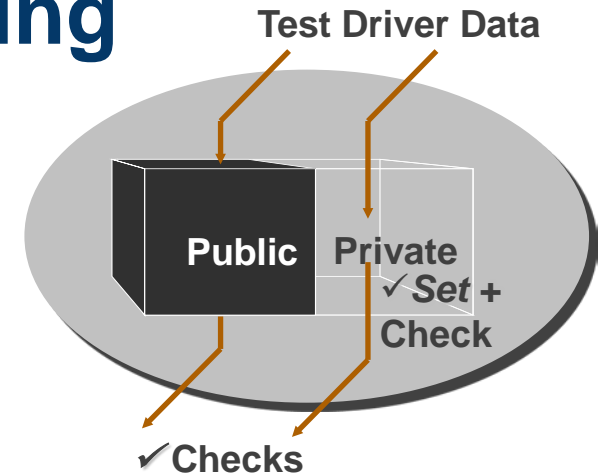
➔ ***Cantata++*** simplifies with...

- Table-driven test generation
 - ☑ Multiple values per parameter
 - ☑ Ranges of values
 - ☑ Functions calculating values
 - ☑ Combinatorial effect calculator
 - ☑ Checks on call sequences and returns
- Robustness rule sets for data types

5 Flexibility: White-Box Testing

Challenges:

- Call private methods/static functions
- Set/check private/static data
- Control of internal calls



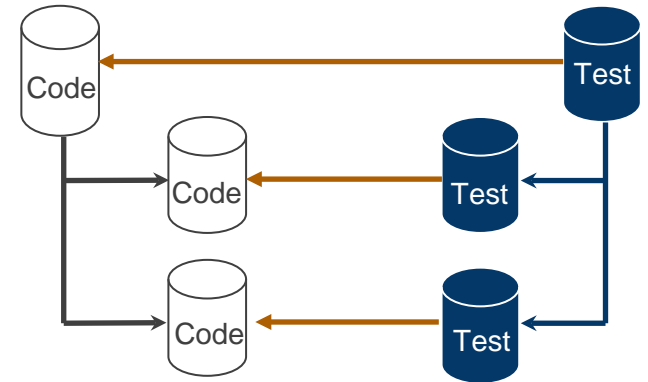
➔ ***Cantata++*** solves the problem through...

- Automated accessibility instrumentation
 - 🔍 Call private methods and static functions directly
 - 🔍 Set/check data which is private, in unnamed namespaces and declared static
- White-box Call Interface Control
 - 🔍 Calls internal to compilation unit
 - 🔍 Wrapping / Isolating OS library calls

5 Flexibility: Object Oriented Testing

Challenges:

- Test case re-use aligned with code
- Support for Templates
- Support for Inheritance
- Testing Abstract Base Classes



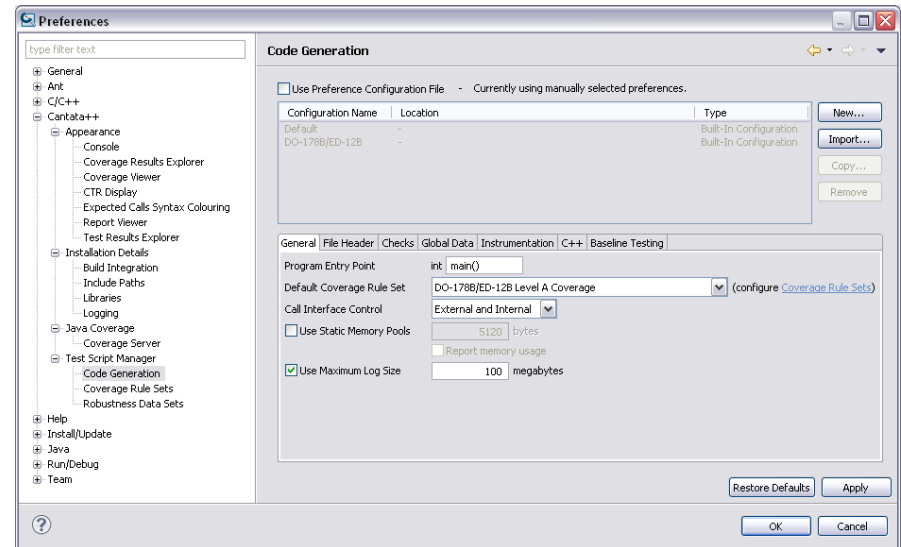
➔ ***Cantata++*** supports these with...

- Parallel hierarchy of code and test case re-use
 - ☑ Template instantiation
 - ☑ Inheritance and factory methods
- Abstract Base Class testing
- Object oriented code coverage
 - ☑ Derived inheritance context
 - ☑ User contexts (State machines or multiple threads)

5 Flexibility: Tool Configuration

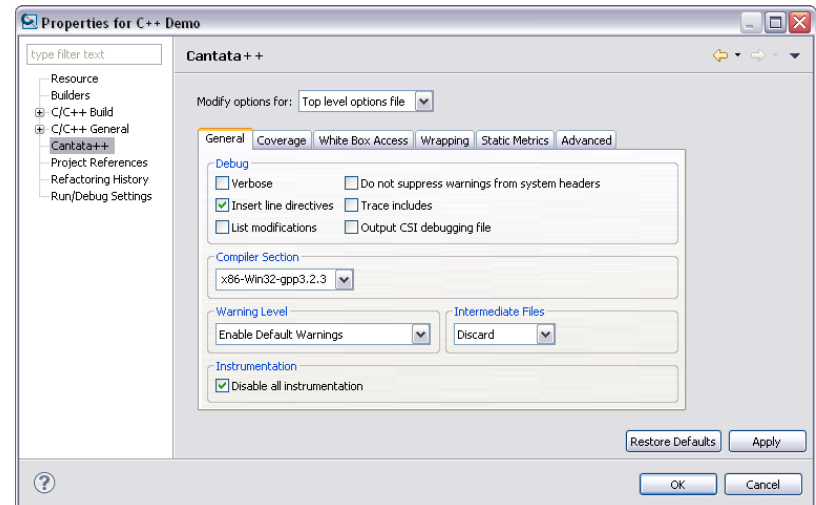
Workspace Tool Preferences

- Stored configurations
- Appearance of views / reports
- Installation details
- Java Coverage
- Test Script generation



Project Specific Properties

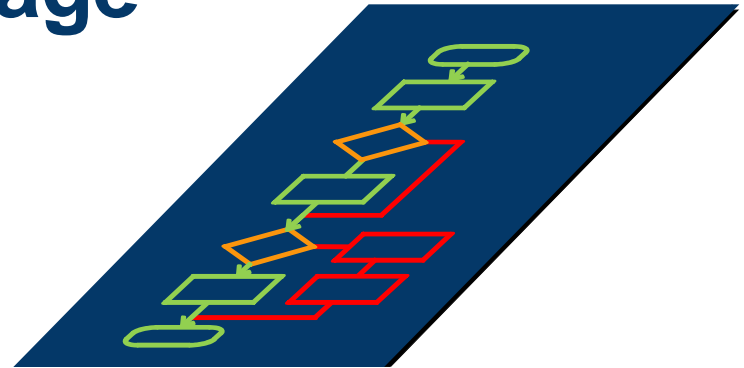
- General & specific tool options
- Advanced options
- Whole project or sub-folders



6 Integrated Code Coverage

Challenges:

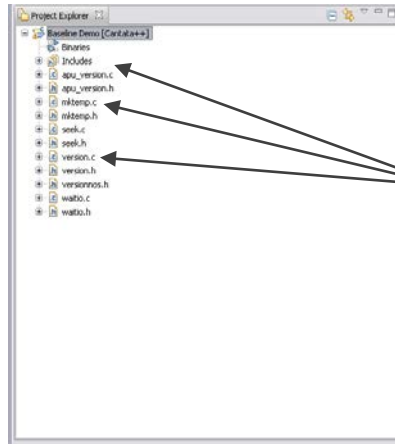
- Set coverage targets in tests
- Diagnostics over stages / test runs
- Sensible coverage metrics
- Coverage redundancy optimisation



➔ ***Cantata++*** advanced capabilities offer...

- DO-178B code coverage rule sets (Levels C to A)
- Powerful drill-down views, filters and reports
- Coverage metrics
 - ☒ Entry- Point Coverage
 - ☒ Statement Coverage
 - ☒ Decision Coverage (as defined by DO-178B)
 - ☒ Call-return Coverage
 - ☒ Condition Coverage (masking & unique cause MC/DC)
- Automatic test case coverage optimisation

Cantata++ Test Generation

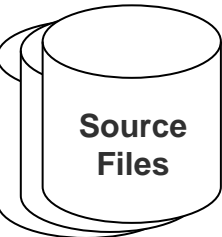


```

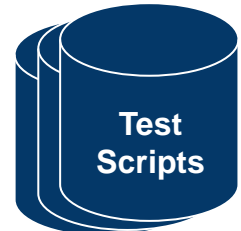
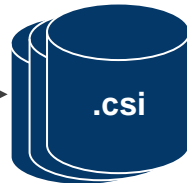
C:\>cantppc - application com.ipl.products.eclipse.cantpp.testscript.BaselineTestGenerator - data C:\my_workspace sourceDirectory=C:\my_source
    
```

Select Source Files(s) to generate Test Scripts

Compile source with **Cantata++**



PARSE



Baseline Test Generation Report







- Generation Preferences**
- Code Coverage Target
 - Data to access (black/white-box)
 - Functions to access (black/white-box)
 - Call Interface Control (stub / wrap / isolate)
 - Baseline Testing:
 - Data to Check (global, parameters, returns)
 - Data to Modify (to force paths)
 - Algorithm limits (time, paths, etc)



***Cantata++* Makefiles & Test Scripts**





***Cantata++* Makefiles**

-  Automatically generated
-  No sandbox copying of source code required
-  Automated CLI invocation
-  Regression Testing









***Cantata++* Test Scripts**





-  Implemented in C / C++
-  Multiple test case types
 - Test per function / method
 - Table Driven
 - Robustness tests
 - Object Oriented testing
 - Baseline testing

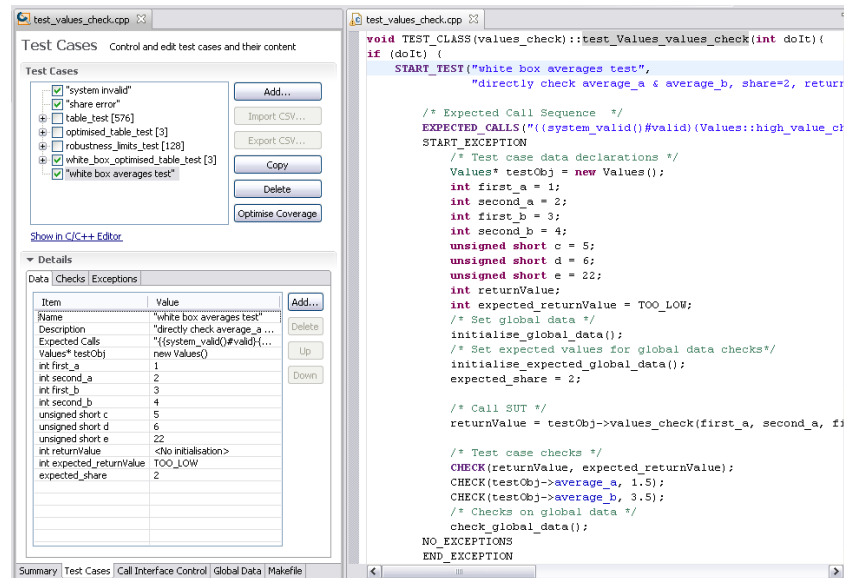
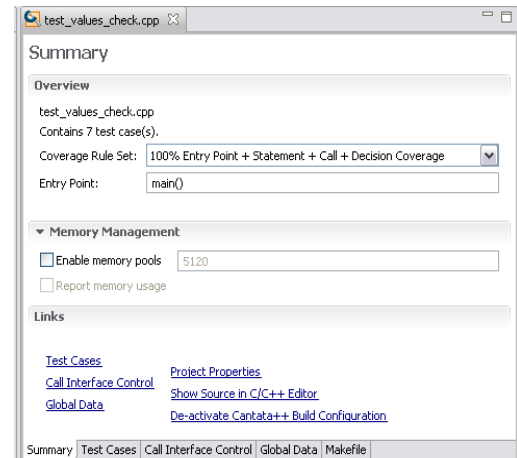
Editing Tests

Cantata++ Test Script Manager

-  Graphical view of C/C++ test script
-  Summary
-  Test Cases
-  Call Interface Control
-  Global Data
-  Makefiles

Synchronized Editors

-  GUI and C/C++ test script
-  Synchronize on save
-  Context sensitive links
-  External editing supported



Cantata++ 6.1 Supported Platforms

Host Operating Systems

- Windows XP, Vista, 7
 - Linux 2.4 & 2.6.x kernel
- 32
+
64
bit

Host Compiler Tool chains

- Microsoft Visual C++ (6.0, 2003, 2005, 2008, 2010)
- GNU gcc/g++ (up to 4.6.x)

Target Platforms

- All Wind River Workbench 3.3 supported platforms
- Any combination of bare metal / RTOS, cross-compiler and processor
- Cantata++ Deploy Target capability for any target platform

IDEs

- Cantata++ supplied Eclipse 3.6.2
- Wind River Workbench 3.3
- IDEs compatible with Eclipse 3.6
- Other proprietary IDEs

Benefits of Intelligent Testing



● The 6 key Technical Tool Challenges

☒ Solved by intelligent solutions

+

● Eases tool adoption

☒ Eclipse IDE & target integration

☒ Tests in C/C++

● Increases efficiency

☒ Automation

☒ Flexible tool-kit

☒ Powerful diagnostics

● Improves professionalism

☒ Structured repeatable testing

☒ Knowing code will be properly tested

Advantages over the Competition



**Integrated with Eclipse
and Target Platforms**

Unique Call Interface Control

**Most Automated
Flexible Testing Tool Kit**

**Most Powerful
Code Coverage Diagnostics**

Automatic Baseline Testing

DO-1768-B Annexes

Process Objectives and Outputs by Software Level

Annex	Title	Objectives supported for levels A-D
A1	Software Planning Process	1, 2, 3, 4, 5, 6, 7
A2	Software Development Processes	1, 2, 3, 4, 5, 6, 7
A3	Verification of Outputs of Software Requirements Process	1, 2, 3, 4, 5, 6, 7
A4	Verification of Outputs of Software Design Process	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
A5	Verification of Outputs of Software Coding & Integration Processes	1, 2, 3, 4, 5, 6, 7
A6	Verification of Outputs of Integration Process	1, 2, 3, 4, 5
A7	Verification of Verification Process Results	1, 2, 3, 4, 5, 6, 7, 8
A8	Software Configuration Management Process	1, 2, 3, 4, 5, 6
A9	Software Quality Assurance Process	1, 2, 3
A10	Certification Liaison Process	1, 2, 3

A-3 Verification of Outputs of Software Requirements Process

6

High level requirements are traceable to system requirements

Including requirements references in Cantata++ test cases

7

Algorithms are accurate

Dynamically testing the correct calculations of the algorithms.

A-4 Verification of Outputs of Software Design Process

3	Low level requirements are compatible with target computer	Dynamically test against requirements on the target computer
4	Low level requirements are verifiable	Dynamically test the requirements
7	Algorithms are accurate	Dynamically test the accuracy of the algorithms.
10	Software architecture is compatible with target computer	Dynamically test architecture compatibility on the target computer
11	Software architecture is verifiable	Dynamically test the architecture
13	Software partitioning integrity is confirmed	Dynamically test partitioning integrity

A-5 Verification of Outputs of Software Coding & Integration Processes

1	Source Code complies with low level requirements	Dynamically test against the requirements
2	Source Code complies with software architecture	Dynamically test against the architecture
3	Source Code is verifiable	Dynamically test the Source Code
4	Source Code conforms to standards	Statically analyse the Source Code for standards
6	Source Code is accurate and consistent	Dynamically test the Source Code
7	Output of software integration process is complete and correct	Dynamically test the integration

A-6 Verification of Outputs of Integration Process

1	Executable Object code complies with high level requirements	Dynamically test against the requirements
2	Executable Object Code is robust with high-level requirements	Dynamically test robustness against the requirements
3	Executable Object code complies with low level requirements	Dynamically test against the requirements
4	Executable Object Code is robust with low-level requirements	Dynamically test robustness against the requirements
5	Executable Object Code is compatible with the target computer	Dynamically test compatibility on the target computer

A-7 Verification of Verification Process Results

1	Test procedures are correct	Reviewable test scripts
2	Test Results are correct and discrepancies explained	Direct from target test results identify passes and discrepancies
3	Test coverage of high level requirements is achieved	Test results include requirements references
4	Test coverage of low level requirements is achieved	Test results include requirements references
5	Test coverage of software structure (modified condition/decision)	Masking and Unique Cause MC/DC coverage reported in test results
6	Test coverage of software structure (decision coverage)	Decision coverage reported in test results
7	Test coverage of software structure (statement coverage)	Statement coverage reported in test results
8	Test coverage of software structure (data & control coupling)	Dynamic checks on data read-write order & call sequence verification

DO-178B/ED-12B Tool Qualification Case



Qualification Requirements and Credentials

Tool Qualification Pack

IPL Quality Involvement

***Cantata++* Tool Development**

***Cantata++* Technical Support**

Further Information

DO-178B

Tool Qualification Requirements

Software verification tools definition

- *“Tools that cannot introduce errors, but may fail to detect them.”*
[DO-178B/ED-12B 12.2]
- **Cantata++** is a software verification tool

Qualification criteria for software verification tools

- *“The qualification criteria for software verification tools should be achieved by demonstration that the tool complies with its Tool Operational Requirements under normal operational conditions.”*
[DO-178B/ED-12B 12.2.2]

Tool Credentials

Cantata

- Original tool developed in 1992 to the IPL QMS

Cantata++

- Development continued in same way from 1998

Audits

Both tools have been successfully audited by customers / DERs a number of times for use on DO-178B/ED-12B level A projects, and for various other standards

***Cantata/Cantata++* audits**

Recent audits of *Cantata/++*

Software Verification Tool for use on DO-178B/ED-12B Level A projects:

- April 2009: audited by TTTech, Hamilton Sundstrand, Boeing
- October 2008: audited by Smiths Aerospace
- January 2007: pre-audit by TTTech
- June 2005: audited by ESW Wedel
- April 2003 and March 2005: audited by Nord Mikro

***Cantata++* Tool Qualification Pack**

***Cantata++* Version Specific CD**

- **Tool Qualification Package Overview**

 - Qualification requirements

 - Qualification process

 - Links to all tool qualification data

 - Advice of product use for DO-178B

CD Availability

- **Provided free of charge under Non-Disclosure Agreement**
- **Available for customers and certification authorities**
- **Support for use of CD in tool qualification provided by IPL**

IPL Quality Involvement



Maintenance of the Quality Management System (QMS)

- ensure QMS meets Company requirements
- ensure QMS meets ISO9001 requirements

Ensure ***Cantata++*** project meets QMS requirements

- Quality involvement throughout software lifecycle
- Projects subject to Quality audit

QMS fits IPL's requirements

Review

- Regular and on-going
- Procedures & Practices

Audits

- Undertaken regularly
- Conducted by Quality team

QMS documentation

- Review and authorisation by Quality Review Board
Quality & Security Manager
Chief Operations Officer
Chief Technology Officer

Scope of the IPL QMS

Applies to all parts of the company including ***Cantata++***

Covers all aspects of the development lifecycle:

- proposals
- requirements
- design, implementation and software integration
- project management
- verification and validation
- configuration management, including change management, version control and build control
- process control, including enhancements
- remedial, corrective and preventive actions
- software maintenance and enhancement
- audits and management review of the QMS

QMS Documentation

***Cantata++* Project Related**

- Software Code of Practice (SCOP)
- Quality Assurance Procedures (QAP)
- Technical Support Procedures (TSP)

Others

- System Management Procedures (SP)
- Training Procedures (TP)
- Consultancy Services Procedures (CSP)
- Management Procedures (MP)
- Sales and Marketing Procedures (SMP)
- Career Development and Monitoring Manual (CDMM)

SCOP Project Documentation

Software Quality Plan (SQP)

- Defines the quality standards

Introduction To Project (ITP)

- Defines the design, development and software production methods
- Defines project specific requirements

How the generic QMS standards and procedures are used

How any QMS standards and procedures are modified or supplemented

Others:

- Project plans
- Progress reports
- Quality records

IPL QMS fits ISO's Requirements

IPL Quality Manual

- Maps IPL QMS to ISO 9001:2008

IPL Quality Function

- Involvement in production/review of new QMS procedures
- Undertakes on-going audit programme

External Assessment

- Regular external assessments of IPL's QMS by qualified ISO/TickIT assessor from LRQA
- Last full re-assessment October 2010 by British Standards Institution

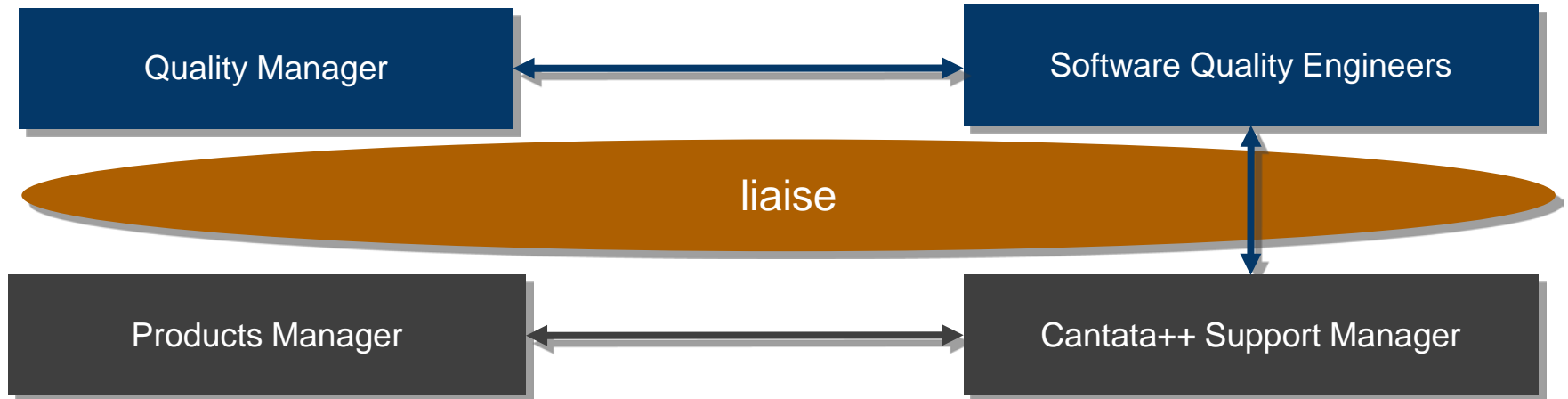
IPL QMS Certification History

- 1981 - IPL's QMS formally documented
- 1988 - ISO9001 certification
- 1989 - AQAP 1 & 13 certification
- 1992 - ISO9001/TickIT certification
- 1995 - ISO9001/TickIT re-certification
- 1999 - ISO9001/TickIT re-certification
- 2001 - ISO9001/TickIT re-certification following transfer to ISO9001:2000
- 2004 - ISO9001/TickIT re-certification
- 2007 - ISO9001/TickIT re-certification
- 2009 - ISO9001/TickIT re-certification following transfer to ISO9001:2008
- 2010 - ISO9001/TickIT re-certification
- 2011 - Will be transitioning to TickITplus



IPL QMS Current Certification
LRQA 4006360

Quality Involvement - Project Lifecycle



Activities

- Proactive project support especially on quality planning
- Review/authorisation of documents
- Project Audits
- Review/witnessing/sign-off of verification and validation
- Maintenance of Quality records

DO-178B Qualification Case Tool Development

chris.bradley@ipl.com



Development Stages of *Cantata++*
Operational Requirements
System Level Testing
Configuration Management

A brief history of *Cantata++*

Cantata++ 1.0 (1998) Initial release



Cantata++ 2.0 (1999) - Static Analysis added



Cantata++ 3.0 (2003) – Test Driver added



Cantata++ 4.0 (2004) – Studio added



Cantata++ 5.0 (2006) – Built on Eclipse



Cantata++ 6.1 (2011) – Current version

Development stages

Original development (to v2.0)

- full development team
- full development/design documentation
- module tests, task tests
- system acceptance tests

Maintenance (post v2.0)

- Fault Reports (FRs) fixed
- minor new requirements – New Feature Requests (NFRs) defined
- small, incremental enhancements Project Change Request (PCRs) implemented
- design documentation, module tests and task tests not maintained
- system acceptance tests maintained
- additional 'extra' tests that test all FRs, PCRs and NFRs added

Operational Requirements

IPL's QMS demands Software Requirements Specification (SRS)

- unambiguous, complete, consistent, correct
- testable
- configuration managed document
- subject to independent and Quality review

For ***Cantata++*** Requirements are defined in the ITP as

- Requirements as they existed at the end of original development (corresponds to Cantata++ v2.0)
- FRs and PCR's done since then (these do not affect the Cantata++ v2.0 Core Component Requirements)
- NFRs done since then (these add new minor requirements)

System Level Testing

The current ***Cantata++*** System Level Tests comprise:

- **System Acceptance Tests (SATs)**

These are essentially the SATs as they existed at the end of original development. They test all the required functionality as of Cantata++ v2.0. The System Acceptance Test Specifications (SATSs), System Acceptance Test Procedures (SATPs) and Verification Cross-Reference Indices (VCRI) from the end of Cantata++ v2.0 apply to these

- **Extra tests for FRs, PCRs and NFRs**

Each FR, PCR and NFR has a separate Extra Test

- **Execution of System Level tests**

SATs and all Extra Tests are executed on the compilers/platform combinations documented in the Cantata++ Software Issue Form for each release.

Configuration Management

All items configuration managed with full traceability

- Documents
- Code
- Tests entities (configuration items) configuration managed

Changes made via formal process

- FRs
- PCRs (enhancements)
- NFRs (new minor requirements)
- Configuration managed reviews of all items modified as part of a change

Library procedures ensure

- Controlled retrieval
- Secure data retention

Tool Qualification Case

Technical Support



Delivery to Distributors / End Users

Products Installation and Deployment

Product Training

Product Support

Product Delivery

IPL Supplies Distributor or Customer

- Initial product installations
- License keys
- Product updates
- DVD or FTP

Documentation

- User Guides and complete reference Manual
- Hyperlinked HTML help and PDF
- Further specific technical notes on Support Website

Installation and Deployment

Installation

- On development Host supported platform

Deployment

- Initial / upgrade deployment of Cantata++ Library
- Any necessary Cantata++ library code changes due to:
 - Compiler restrictions
 - Compiler bugs

Verification of correct behaviour:

- Automated Cantata++ Deployment Tests
- Deployment Port Report for each precise target configuration

Product Training

Training available

- Classroom product use training courses

Significantly increases the productivity of development and test engineers performing unit and integration level testing of C/C++.

Hands-on workshop covers the use of the Cantata++ product with examples and users own code.

- Product orientation webinars
- On-site product use consultancy and mentoring

The image displays a collage of training materials for Cantata++. At the top left is the Cantata++ logo with the tagline 'Intelligent testing'. Below it is a 'Workshop Details' table:

Course Title	Cantata++
Duration	2 Days
Format	Instructor-led lectures and hands-on lab exercises
Price	Contact your supplier

Below the table is the 'Workshop Description' section, followed by 'Products Supported' (Cantata++ 5.0 and 6.0), 'Who Should Attend' (Development and Quality Assurance/Quality Control personnel), 'Prerequisites' (Basic understanding of C/C++ programming), and 'Course Format' (Hands-on workshop).

The main part of the collage is a 'Training Workshop' agenda for 10th-11th January 2011. It lists 'Topics Covered' such as 'Intelligent testing', 'Requirements', and 'Contact Us'. It also includes an 'Agenda' section with 'Day 1' and 'Day 2' topics, and a 'Training Workshop Instructor' section.

Licensed Product Support

Customer Data

- Customer details
- Product delivery and installation and technical use environment
- Any customer supplied code
- All data securely held

Customer Enquiry

- All product use enquiries replied to within 1 working day
- Customer queries - Customer Enquiry Problem Report (CEPR)
 - Raised if not resolved in first contact
 - May result in Fault Report (FR), Project Change Request (PCR) or workaround
 - CEPRs status monitored
 - Any faults affecting validity of results reported to all registered users

The screenshot displays the 'E Honeywell' software interface for a Customer Enquiry Problem Report (CEPR). The form includes the following fields and sections:

- Company:** Honeywell
- Site:** BANGALORE
- CEPR Number:** 1419
- Contact:** Ms S Giri
- Status:** Closed
- Lic No:** CTA0356
- Date/Time Opened:** 03/09/2007 16:06:11
- By:** toml
- Product:** Cantata++
- Schedule:** #
- Date/Time Pending:**
- By:**
- Date/Time Closed:** 20/12/2007 14:27:49
- By:** phild
- Category:** Fault, Enquiry, Enhancement No.
- Affects:** Component: Canata++, Platform: PC, OIS: Windows, Version: 5.1.3, Severity: Medium, Compiler: n/a
- Summary:** CPPCCD doesn't allow wild cards
- Description:** toml: They want to use wildcards in their CPPCCD commands, but it seems this is not possible on Windows
- Outcome:** Fixed in 5.2, customer informed.
- Buttons:** First, Prev, CEPR 798 of 818, Next, Last, Paste Details, Send To Clipboard, Cancel, OK

Cantata++ for DO-178B

Further Information

- Civil and Military Avionics Sector Briefs
- Customer Case Studies
- DO-178B Tool Qualification Package CD



Case Study: Vibro-Meter

Critical Avionics

Critical Failure Testing Requirements

Recently, Vibro-Meter has been analyzing two major products: the Engine Inverter and Power Monitoring Unit (EMI) for the Airbus A320 aircraft and the Engine Monitoring Unit (EMU) for the Airbus A320neo aircraft. One of the main challenges of these developments was to make reliable software that had been fully tested under stringent conditions of the old avionics to be replaced.

The full range of the former systems and IT systems and methods of such software, is widely inherited by a new system in a similar manner. Level 0 with accident, and so on.

Seven Years of Software Testing of Vibro-Meter

Vibro-Meter is one of the world's leading providers of vibration testing systems, used in a wide range of applications, including aircraft, aerospace, industrial and other non-aerospace applications. In 2007, Vibro-Meter was acquired by Intellicore, a subsidiary of Intellicore PLC, which was then acquired by Intellicore PLC in 2012. Intellicore PLC is a leading provider of software measurement and analysis systems for engine testing, including the development of the world's major engine manufacturers (GE, Honeywell, Pratt & Whitney, etc.).

AIMS - Program B

Vibro-Meter has been the primary provider of software testing systems for the AIMS Program B, which is a major program for the development of the A320neo aircraft.

Helicopters

Helicopters: Airbus Helicopters, Sikorsky, etc.

Equipment

Equipment: Intellicore, etc.

Non-Fighter Aircraft

Non-Fighter Aircraft: Airbus A320neo, etc.

IPL Cantata++ AdaTEST 95

Key Standards

- DO-178B (ED-128)
- DO-178C (ED-129)
- DO-178D (ED-130)

Tool Qualification

IPL's testing tools have been qualified in accordance with the requirements of the DO-178B (ED-128) standard.

Technologies

Cantata++ and AdaTEST 95 are qualified in accordance with the requirements of the DO-178B (ED-128) standard.

Development Tools

Development Tools: Intellicore, etc.

IPL Cantata++ AdaTEST 95

Civil Avionics Sector Brief

Key Standard

- DO-178B (ED-128)

Tool Qualification

IPL's testing tools have been qualified in accordance with the requirements of the DO-178B (ED-128) standard.

Technologies

Cantata++ and AdaTEST 95 are qualified in accordance with the requirements of the DO-178B (ED-128) standard.

Development Tools

Development Tools: Intellicore, etc.

Other Avionics

Other Avionics: Intellicore, etc.

Military Avionics Sector Brief

Key Standard

- DO-178B (ED-128)

Tool Qualification

IPL's testing tools have been qualified in accordance with the requirements of the DO-178B (ED-128) standard.

Technologies

Cantata++ and AdaTEST 95 are qualified in accordance with the requirements of the DO-178B (ED-128) standard.

Development Tools

Development Tools: Intellicore, etc.

IPL Information Processing Limited

Tool Qualification Package Overview

TQPO for Cantata++ 6.0

Project Reference: [Redacted]
Document Reference: [Redacted]
Date: 14 Jan 2011
Issue: 4

999185 TOPO 14 Jan 2011 4

Page 1 of 35

IPL