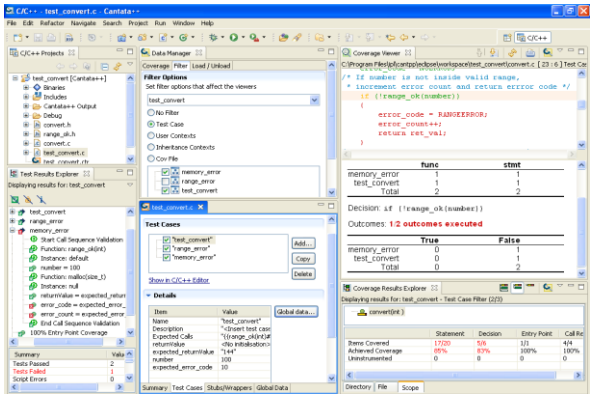
BUILT ON eclipse

Cantata++ 6.1

intelligent testing

Cantata++ provides advanced high productivity techniques, allowing developers to dynamically prove their code with intelligent unit and integration testing, in the most cost effective manner.



The Cantata++ Eclipse user interface provides a complete test development environment for the creation, execution and analysis of unit and integration tests.

It easily integrates with the developer desktop, compilers and embedded target platforms.

Why Unit/Integration Test

Standards Compliance

International standards for most safety related industries require that evidence of low level testing and how much of the code was tested, is provided to obtain certification for the device software.

In addition most standards require or provide certification credit for the test tool to be qualified / certified or proven through use.

Cantata++ has been successfully used by customers worldwide since the 1990s to meet all the international safety related standards including; IEC61508 and its various sector specific derivatives, DO-178B for avionics, and IEC62304 / FDA Guidelines for medical devices.

Commercial Risk

Thorough software testing, only possible at the developer level due to application complexity, is necessary to ensure that shipping devices:

- perform reliably
- do not negatively impact brand or corporate reputations
- provide a defence against litigation on grounds of fitness for purpose

Cost of Testing

Unit testing and low level integration testing are the earliest tests performed during software development.

This level of testing is also the least expensive time to fix software bugs, and can be done before bugs pollute the wider code stream, or affect shipping devices.

Intelligent Testing Benefits

Meets Key Technical Tool Challenges

- **Planning Test Effort**
  Knowing where to focus my test effort and how much work it will be
- **Tool Reliability**
  Not fighting the tool, but working seamlessly with what I already have
- **Maximising Test Automation**
  Finding time to create reusable test harnesses to test all my functionality
- **Call Interface Control**
  Making it simple for me to test object interactions with the rest of the system
- **Flexible Test Techniques**
  Having a full tool-kit to efficiently test the way I need to
- **Code Coverage**
  Confidence that my tests on the target cover the important code
- **Testing Legacy Code**
  Needing to create unit regression tests for my existing code to safely modify it

Eases Tool Adoption

Eclipse IDE and target integrations, reduce adoption barriers and learning curve

Increases Efficiency

Intelligent techniques, applied with powerful diagnostics focus effort only where needed

Improves Developer Professionalism

Structured, repeatable thorough testing imposes coding discipline on developers

Cantata++ prove your code with intelligent unit testing

Product Overview

Valid from September 2011

Free Trial

Free trial of Cantata++ on your host or embedded target platform

REQUEST NOW

Or contact your local supplier.

Major Features

- Unit and Integration testing
  - C and C++
  - Development host and embedded target platforms
- Highly automated developer testing reducing cost and increasing quality
  - Re-usable test harness
  - Test script generation written in C/C++
  - Unique call interface control to simulate and intercept calls
  - Automatic white-box accessibility
  - Large data sets and robustness testing
  - Automated regression testing
- Automatic generation of complete baseline of unit tests for legacy C code
- Most advanced integrated code coverage analysis available for C/C++ and Java
- Fully qualifiable for safety critical use

Contact Us

nohau

- Web nohau.se
- Email info@nohau.se
- Tel +46 (0) 40 59 22 00

Copyright © IPL 2011.
All trademarks acknowledged.
Cantata is a registered trademark of IPL Information Processing Limited

Quality Pedigree

Cantata++ has been developed under the control of IPL's Quality Management System which is certified to ISO 9001:2008 (TickIT).

Certificate LRQ 4006360

Cantata++ 6.1 is the latest version of the IPL Cantata suite first released in 1992.

Cantata++ has been proven through use in all high integrity sectors including: medical, air-traffic control, avionics, rail transport, automotive, space, defence, nuclear energy; as well as for business critical sectors such as telecoms and financial and digital broadcast systems.

See customer case studies and Sector Briefs, for more information how customers have proven their code with intelligent unit testing using Cantata++.

Tool Qualification

A Cantata++ Tool Qualification Package is available free of charge for customers working to DO-178B, IEC 6150-8 and other high integrity standards.

Training & Support

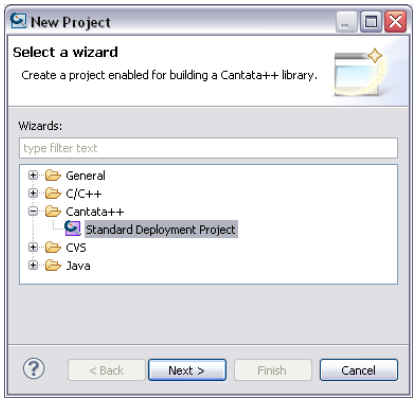
On site Cantata++ training courses, product consultancy and technical support are available through an extensive network of distributors and IPL.

Test Planning

- **Static Analysis** generates over 300 source code metrics (procedural and object oriented) in csv format. The metrics for maintainability and complexity aid estimating test effort.
- **Requirements Based Testing** is supported through tagging of test cases for requirements traceability into reports.

Test Automation

- **Platform Deployment** of Cantata++ to the user's specific host or embedded target build environment by end users through a deployment wizard.



- **Test Script Generation** by parsing source code to derive parameter, calls and data information with call interface control (Stubs, Isolates and Wrappers) generated into the test script(s).
- **Test Script Manager** provides graphical development, editing and management of tests synchronised with the C/C++ editor.
- **Re-use of xUnit Tests** within Cantata++ tests, retains investment in open source scripts, allowing them to be re-used in and extended with Cantata++ features.
- **Intuitive Test Directives** reduce manual scripting, by calling library of helper functions to provide structured, readable, independent, repeatable tests with clear and unambiguous results.

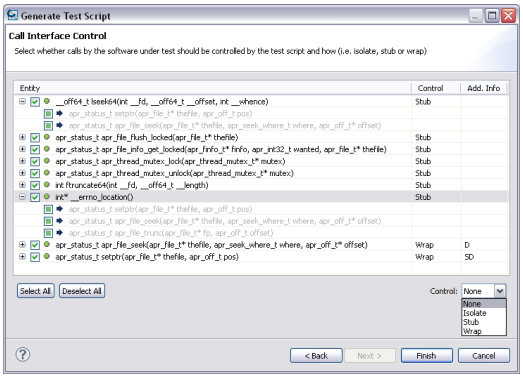
Cantata++ Makefiles

- **Automatically Generated** when tests are created, Cantata++ Makefiles compile, link, deploy, execute and retrieve results for suites of tests in batch mode.
- **Used With Existing Code Tree** allows multiple test scripts to reside in the same Eclipse project, and supports automated regression testing and Continuous Integration (CI) via both UI and CLI.
- **Easily Integrated** with open source or commercial CI tools, Cantata++ Makefiles can also be edited via the Test Script Manager UI.

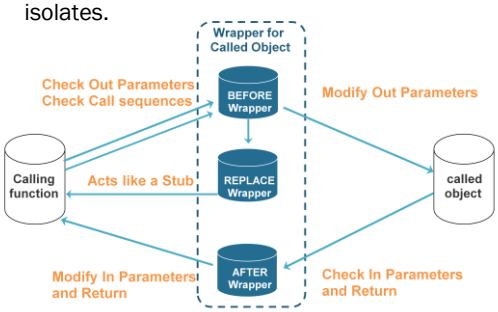
Call Interface Control

All calls (external and internal to the compilation unit boundary) may be simulated by Stubs / Isolates or intercepted using the real call via Wrappers.

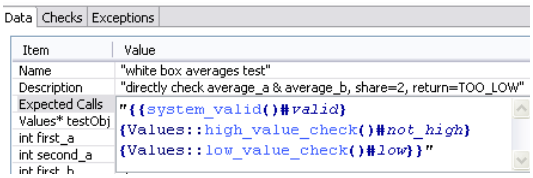
- **Interface Control Selection** for method used on each call at test script creation (with defaults), and script editing, with call list is refreshed on each code build.



- **Optional Automatic Checks** of all parameters / returns passed over call interfaces.
- **Programmable Test Control Points** are available at each call interface control through named instances to set and check data, raise exceptions etc
- **Automated Stubs** provide replacement simulation of called object interface.
- **Automated Isolates** provide simulation of linked called object interface, and support system and variadic functions.
- **Automated Wrappers** provide interception of linked called object interface with before/after and before/replace modes to intercept or simulate. More powerful and flexible control over interfaces than stubs or isolates.



- **Call Sequence Verification** provide full control over call order and use of Stub, Isolate or Wrapper instances in each test case, with exact sequence or any time matches.



Cantata++ prove your code

Black & White-Box Testing

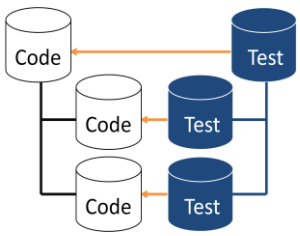
Cantata++ provides a high degree of automation at unit and integration levels for both black-box and more efficient / thorough white-box testing.

- **Per Function Testing** generates a complete template test case for each function/method in the code with all parameters, accessible data and calls to control identified,
- **Table-driven Testing** provides multiple input value ranges, combinatorial effect calculator and CSV import/export for large data set black-box verification.
- **Robustness Testing** with pre-defined values for basic data types, in table driven test cases.
- **Automated Global Data Checks** to verify expected and unexpected changes to all global data accessible to the software under test.
- **Automatic White-Box Access** provides the ability to efficiently call functions or methods, and set/check data directly from the test script without conditional compilation:
  - Static functions and private methods
  - Private data and data static to the file
- **Automated C++ Exception Verification** of multiple expected and unexpected exceptions.

Object Oriented Testing

Cantata++ unit and integration testing supports: C++ functions, templates, classes or clusters of classes in single or multiple source files. Test scripts for classes are written in C++ as a test class.

- **Automated Test Case Re-use** via parallel inheritance hierarchy, retains benefit of code re-use for testing, and verify that a derived/specialised class has the same semantics as a base or template class (the Liskov Substitution Principle).



Parallel inheritance hierarchy

- **Automatic Test Class Inheritance** structure created for inherited classes, so child class tests inherit from parents.
- **Automatic Implementation** of abstract base classes (ABCs) or pure virtual methods, with generation of appropriate stubs in the script. Supports use of the factory method.

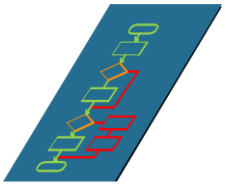
with intelligent unit testing

Code Coverage

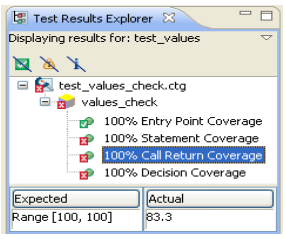
Coverage analysis of C/C++ and Java provides objective measurement of how effective testing has been in executing the source code.

- **Code Coverage Metrics** supported include:
  - Entry points
  - Call Returns*
  - Statements
  - Basic Blocks*
  - Decisions (branches)
  - Conditions*
  - MC/DC (Masking and Unique Cause) *

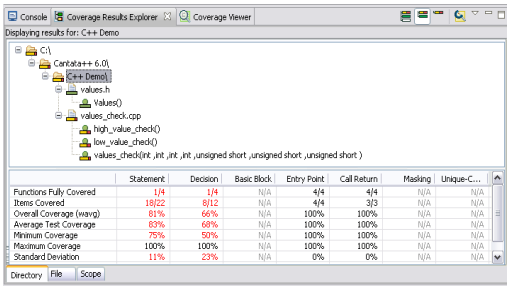
  *not supported for Java



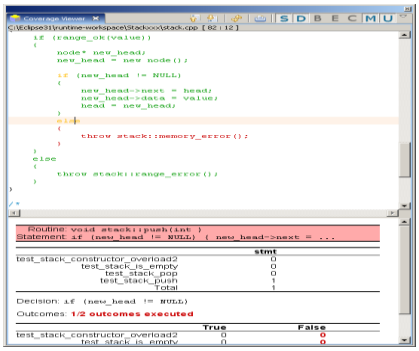
- **Coverage by Context** provides coverage by
  - Each Cantata++ Test case
  - Derived inheritance contexts
  - User defined context for multiple states, threads and data contexts (for DO-178B data coupling analysis)
  - Any non-Cantata++ test run
- **Configurable Coverage Requirements** are easily defined in simple Rule Sets.
- **Coverage Target Checks** on metrics are integrated into dynamic tests resulting in Pass/Fail for coverage requirements.



- **Automatic Test Case Optimisation** provides option to remove/disable test cases which do not contribute additional coverage.
- **Project Code Coverage Trees** for multiple projects with filters for coverage types / fully covered code items and drill-down to highlighted source code.



- **Source Code Coverage Views** highlight individual code constructs (not just by line) with additional diagnostics by test case, test run and metric type.



Flexible Use

Cantata++ provides a flexible tool kit for C/C++ developers to perform unit and integration testing on host or embedded target platforms more efficiently.

- Static analysis metrics aid planning most efficient tests
- Powerful in-built EDG C/C++ parser generates detailed understanding of code, for automatic generation of test cases as thorough or light touch as required
- Preferences for the appearance of all views, reports and installation details can be saved in user specific configurations.
- Test and Analysis Generation options for test script layout, source language, memory, global data and testability instrumentation can all be user configured.
- Test harness automates testing as appropriate for:
  - black-box / white-box
  - positive / negative testing
  - requirements / robustness
  - single / large input data sets
  - procedural / object oriented
  - call simulation / interception
  - isolation / integration test
  - host / target execution
  - requirements / baseline test
- Unique testability instrumentation and wrapping provide precision test control and verification
- Integrated code coverage and test optimisation pin-points gaps and efficiently directs test effort
- Results and diagnostics used for rapid error identification and to support certification
- Automated repeatable execution through Cantata++ Makefiles retains value of tests whenever code is built

Platform Availability

Host Platforms

Windows XP, Vista, 7
Linux 2.4 & 2.6 Kernel
GNU GCC/g++ up to 4.6.x
Microsoft Visual C++ up to 2010
Intel C++ 12
Eclipse 3.6 (if used as a plug-in)

Embedded Target Platforms

IPL Cantata++ can be deployed on almost any target platforms on request.

Wind River Platforms

Through a very close partnership integration, Cantata++ 6.1 is optimised for:

- Wind River Workbench 3.3.1
- VxWorks 5.5 and 6.3-6.9
- VxWorks DO-178B and 61508 (6.6 Cert)
- VxWorks 653 2.3
- Wind River Linux 3.0, 4.0, 4.1, 4.2
- Earlier versions of Wind River Workbench are supported by earlier versions of Cantata++

Further Information

Available at ipl.com/cantata++ from your local supplier or the direct links below:

- Request an online demo
- Latest News from IPL
- Detailed Feature Briefs on specific capabilities
- Sector Briefs documenting customers, standards and technologies where Cantata++ has been used
- White Papers covering management and technical issues and standards compliance
- Customer Case Studies by sector
- Cantata++ tool qualification package

Baseline Test Legacy Code

Cantata++ Baseline Testing automatically generates complete passing unit test scripts to:

- Reduce reliance on system tests
- Support Continuous Integration
- Automatically close gaps in coverage
- Identify testability problems in code
- Easily change unit testing tools

Configurable Test Depth with code paths determined by selecting the metric types in a code coverage Rule Set.

Configurable Automatic Verification using standard workspace preferences for passing checks on function(s) in each file of:

- Return value from function
- Output parameters from function
- Accessible global data values
- Order of function calls made
- Parameter values to function calls

IDE Test Generation for selected source files or directly into existing test scripts.

CLI Test Generation for larger code bases selected by source directories, files and functions.

Separate Standard Licence Feature allows controlled use of Baseline Testing

Testability Issues Identified with warning messages for problems in code preventing generation of complete passing tests:

- Dynamically unreachable code
- Crash scenarios
- Data uninitialised or function static
- Implicit function declarations
- Compiler type truncation

Baseline Test Generation Report created as HTML report for fully, partially or untested files and functions.

Automatic Regression Suite of Baseline Tests created using Cantata++ Makefiles for CLI invocation.

Ongoing Maintainability of Baseline is simplified as test scripts are standard Cantata++ style, with re-use of call instances and detailed path solving description for each generated test case.

Diagnostics & Reports

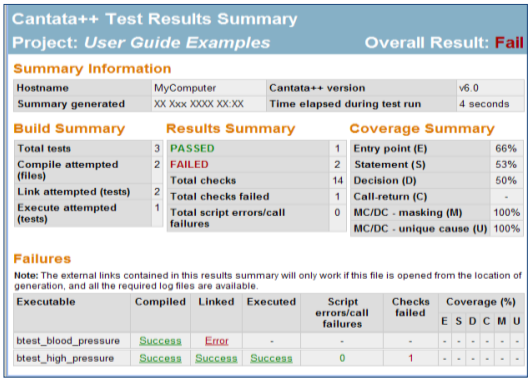
Cantata++ offers powerful filterable diagnostics of test and code coverage results within the UI, and flexible user configurable reports.

Detailed Test Diagnostics for all checks of expected against actual results by test case for:

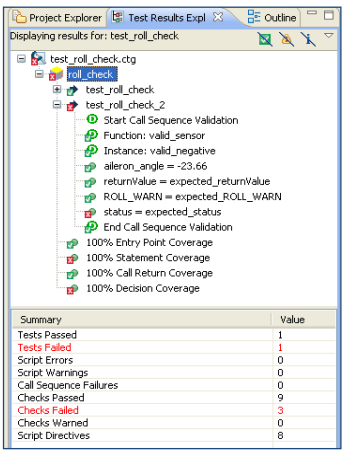
- Global data
- Parameters and returns
- Exceptions
- Call order
- Code coverage targets

Printable Views from within Cantata++ UI for all test and coverage results displays.

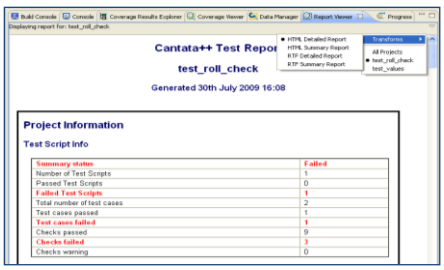
Test Summary Reports for test build and execution results of all tests with Makefiles.



Project Level Tree Views of test pass/fail results with drill-down hyperlinked navigation to individual test cases and checks.



Configurable XML reports with summary or full test details consolidate across multiple projects.



ASCII Text Report available with passes and fails highlighted inside IPL Cantata++ with outline view, or as plain text results file for high integrity certification needs.

Cantata++ prove your code with intelligent unit testing