CANTATA

What's New in Cantata 9.0?

Cantata 9.0, available from October 2019, is a major new release providing significant new functionality including support for Test Driven Development (TDD) and AutoTest support for C++.

This document details the most important changes in Cantata version 9.0.

© Copyright QA Systems GmbH 2019

Introduction

Cantata 9.0, available from October 2019, is a major new release providing significant new functionality. This version introduces support for Test Driven Development (TDD) and enhances Cantata's cutting edge AutoTest capability to provide automatic test generation for C++ code.

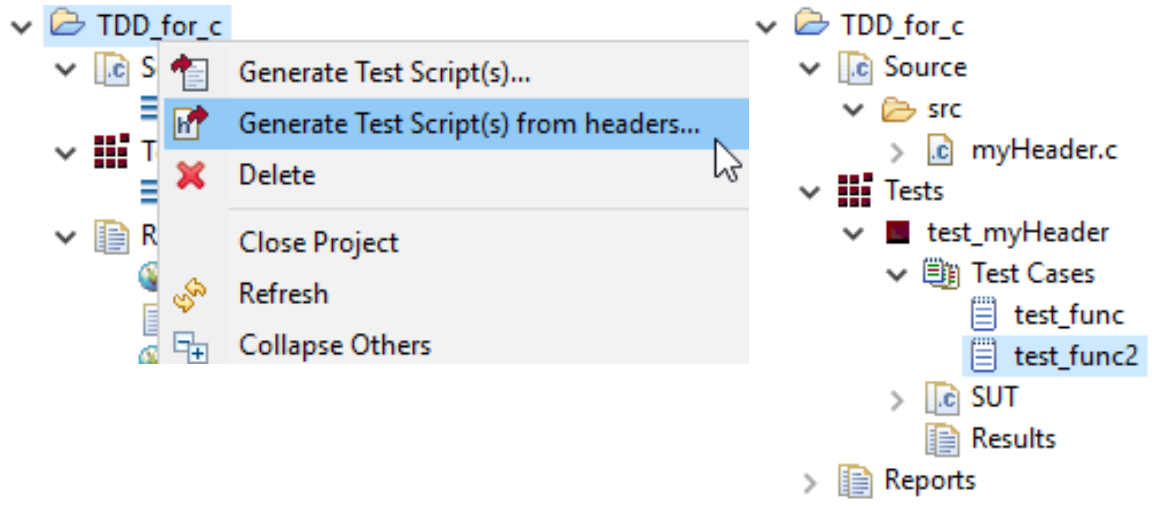
Cantata 9.0 also contains many other smaller enhancements and fixes. The full set of changes are documented in the Release Notes which track all changes in Cantata since version 4.1. The most important changes are highlighted in the sections below.

Test Driven Development Support

Support for Test Driven Development (TDD) has been added to Cantata. TDD is a development method where unit tests are written before and against which source code is implemented. This agile technique can be used to help write "cleaner" code by ensuring that the focus is on development of test cases from requirements, rather than starting with source code implementation.

Cantata 9.0 has new features to support TDD and to make it easy to produce tests for source code which has not yet been written.

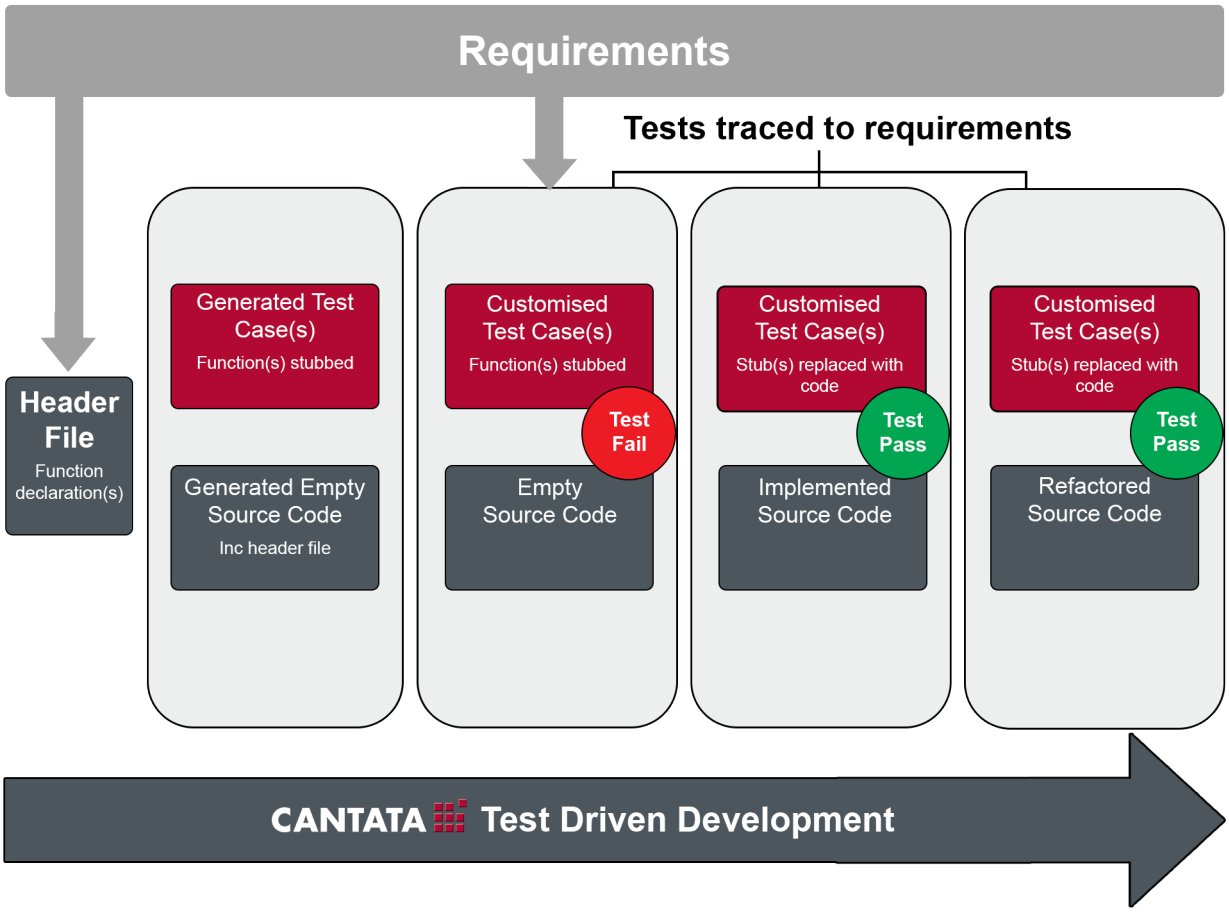
The TDD function can be used to generate test cases as soon as function prototypes are created within header files. A standard Cantata project is created, providing the test framework and fully certifiable reporting one would expect from Cantata. Using Cantata for TDD improves on the simple black-box testing usually used in TDD, giving access to full featured white-box testing as Cantata can directly call encapsulated code internals such as private / static data and functions.



TDD generation in Cantata Test Explorer view

© Copyright QA Systems GmbH 2019

Test scripts are automatically generated with a test case per function prototype defined in the header file. These can be used as the basis for creating further test cases, avoiding the need to manually add information contained in the function prototype to the test cases. Any global data declared in the header file is also automatically set and checked. Cantata TDD generates empty source files, so that once the function bodies are implemented, the tests can simply be rebuilt and run to test the new code.



The Cantata Trace feature can be used in combination with Cantata TDD to improve development of test cases from requirements. Imported Requirements can be traced to test cases as they are designed. Links between test cases, code and requirements provide clarity and make it easier to refactor code later.

Stubs are generated for all functions where the body has not yet been written, this enables the test cases to run before code has been fully implemented. These tests will fail until the correct implementation has been added. Once code has been implemented, these stubs can be removed or converted to Cantata wrappers (to intercept and continue to check calls to these objects/functions) automatically using Cantata Code Change Analysis. Once tests pass, the source code can be refactored and tests re-run to ensure that they still pass.

© Copyright QA Systems GmbH 2019

AutoTest for C++ Code

The Cantata AutoTest feature has been extended to support C++ code. AutoTest automatically generates passing Cantata unit tests to exercise C++ source code, which can be used on new or legacy projects making it easy to:

- Configure automatic test generation
  - choose how thorough test path generation should be, via a selected code coverage ruleset
  - configure the style of the test, through test generation options for stubs/wrapping, checks on data, etc.
- Generate test scripts for a full suite of unit test scripts with 100% code coverage
- Plug 'edge case' gaps in code coverage left after requirements-based test cases by adding AutoTest generated test cases for uncovered functions / methods to existing Cantata test scripts
- Create a thorough safety net of baseline regression unit tests
- Upgrade from other test tools to Cantata

AutoTest parses C++ source code to determine all possible paths through the code as defined by a structural code coverage metric target (i.e. 100% function Entry-points, 100% Statements, 100% Decisions, or 100% Unique Cause MC/DC). An algorithm creates test case vectors that exercise all required code paths, using powerful Cantata white-box capabilities to set data, parameters and control function call interfaces.

Cantata Test Results Summary

Project: *AutoTest Multi File Demo* **Overall Result: Pass**

Summary Information

Hostname	Matt	**Cantata version**	v9.0
Summary generated	Jan 17, 2019, 11:30 AM	**Time elapsed during test run**	16 seconds

Build Summary

Total tests	4
Compile attempted (files)	8
Link attempted (tests)	4
Execute attempted (tests)	4

Results Summary

PASSED	4
FAILED	0
Total checks	233
Total checks failed	0
Total script errors/call failures	0

Coverage Summary

Entry point (E)	100%
Statement (S)	100%
Decision (D)	100%
Call-return (C)	100%
MC/DC - masking (M)	100%
MC/DC - unique cause (U)	100%

The test vectors drive the code, and check the parameters passed between methods, values of accessible global data, order of calls and return values. AutoTest generated cases are editable in the same ways as user generated cases. Each test case has a description of the path through the code it was created to exercise, making them easy to maintain and trace to requirements.

© Copyright QA Systems GmbH 2019

AutoTest in Cantata 9.0 supports the following elements for C++ 03 and earlier C++ versions:

- C++ concrete & abstract base classes
- Overloading and inheritance
- Namespaces and classes
- Exception handling
- Templates when explicitly instantiated within the given code
- Mixed C & C++ code bases

Cantata AutoTest can be used to test C++ code containing elements/versions other than the ones listed above. Tests for supported C++ code will be auto-generated, and additional tests can then be added to achieve the required code coverage target.

Code Coverage Enhancements

Build Variant Coverage

Build Variant Coverage has been added to improve C/C++ coverage data for source code executed over more than one build variant. To do this Cantata uses pre-compile defines (#defines) to separately identify the different build variants. Cantata Coverage Viewer can now show aggregated data for multiple build variants, of the same source code. A report can also be generated with aggregate coverage data across all defines, which is suitable as certification evidence for all build variants of the source code.

```
#ifdef ALARM_PRESENT
    if (alarm_needed) {
        sound_alarm();
    }
#else
    write_status(status);
#endif
    return status;
}
```

Above: Coverage Viewer Showing Code Executed When Build Variant Does Not Include 'Alarm Present'

Right: Example Build Variant Report

```
================================================================================
= Cantata Test Harness v9.0                                                    =
= (c) 2019 QA Systems GmbH                                                     =
=-------------------------------------------------------------------------------=
= Cantata Build Variant Report                                                 =
================================================================================

Input Preprocessor Log Files:
/home/liam/test_cpp_preproc/tmptest/expected_test_file_1.cpl

------------------------- File: 1. test_file_1.c
Preprocessor Directives:
test_file_1.c (7) else
test_file_1.c (11) ifdef DEF_2 >> NOT INCLUDED
test_file_1.c (13) else
test_file_1.c (17) ifndef DEF_4
test_file_1.c (19) else >> NOT INCLUDED
...
------------------------- File: 2. test_file_2.c
Preprocessor Directives:
test_file_2.c (5) if DEF_0 >> NOT INCLUDED

================================================================================
= File                                  Included  Not Included    RESULT =
=-------------------------------------------------------------------------------=
= 1. test_file_1.c                            12            26   >> FAIL =
= 2. test_file_2.c                             0             1   >> FAIL =
=-------------------------------------------------------------------------------=
= TOTALS                                      12            27   >> FAIL =
================================================================================
```


© Copyright QA Systems GmbH 2019

Deferred Coverage Analysis Reports are Certifiable

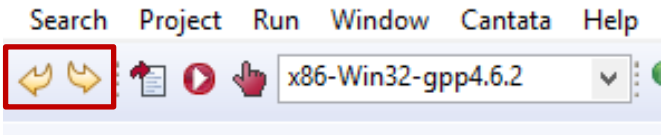
Cantata deferred analysis passes coverage data gathered on target back to the host machine before running coverage checks and reporting. This is useful for targets with a smaller amount of memory. The Cantata CPPGETCOV utility program is now included in the tool certification from SGS-TÜV SAAR GmbH. This means that the .ctr reports generated using this method can now be used as certification evidence.

HTML Coverage Files

Coverage results files can now be generated in an HTML format. These generated files contain project data normally displayed in the Cantata Coverage Viewer.

Undo & Redo

All changes made to test scripts within the Cantata specific views (i.e. Test Explorer, Test Case Editor, or Test Properties) are now automatically saved. Undo/redo functionality has been added for Cantata views, so that these automatic saves can be undone easily. The existing Eclipse buttons can be used to undo and redo edits within Cantata, alternatively ctrl+Z (undo) or ctrl+Y (redo) keyboard shortcuts can be used.



Extended Platform Support

As with every version of Cantata, support for platforms has been extended.

Cantata is tightly integrated with leading Integrated Development Environments which are Built-on-Eclipse®, and toolchains available as Eclipse-Ready® plug-ins. Cantata 9.0 is built on the Eclipse 2018-09 release (Eclipse 4.9), and is also available to install as an Eclipse-Ready plug-in set for Juno (4.2) up to Eclipse 2018-09 (4.9), giving instant access to the ecosystem's latest rich set of plug-and-play tool integrations (e.g. ALM, CI and SCM tools).

Support for the GNU GCC and g++ compilers has been extended to version 8.2 on Windows, and 8.3 on Linux.

Cantata 9.0 has been enhanced to fully support all C++ 11 & 14 language features. For a full list of C++ 11 and C++ 14 language features with hyperlinks to their open-std.org detailed definitions, please contact sales@qa-systems.com

© Copyright QA Systems GmbH 2019