Vector Software

WHITEPAPER

Using VectorCAST to Automate Testing Across the Software Testing Lifecycle

VectorCAST Product Family

The VectorCAST family of products automates testing activities across the software development lifecycle. The purpose of this paper is to provide a high-level view of what the VectorCAST products do, how they work together, and how you might best use them to fully meet your own software testing needs.

The VectorCAST product family consists of five complimentary tools:

VectorCAST/C++ VectorCAST/Ada	*These tools (collectively VectorCAST/C++ and VectorCAST/Ada) automate the process of testing source modules written in C/C++ or Ada/Ada95. They automate both unit testing and integration testing and show the associated code coverage.*
VectorCAST/Cover	*This tool performs code-coverage analysis during functional or systems test. It can share coverage data with VectorCAST/C++ (VectorCAST/Ada).*
VectorCAST/RSP (Runtime Support Package)	*This tool serves as an extension to VectorCAST/C++ (VectorCAST/Ada). It allows you to execute test cases for real-time applications in an embedded-target or simulator environment.*
VectorCAST/Manage	*This tool serves as an extension to the VectorCAST family of unit and integration testing tools. It allows you to import previously developed VectorCAST/C++ (VectorCAST/Ada) test environments into regression test suites, providing a single point-of-control for all unit and integration regression test activities.*
VectorCAST/Requirements Gateway	*This tool permits the flow of data between a requirements management tool and the VectorCAST testing tools. Through a simple and intuitive interface, developers can quickly and easily link requirements to VectorCAST test cases.*

Figure 1: High level view of VectorCAST products

VectorCAST/C++ and VectorCAST/Ada

VectorCAST/C++ (VectorCAST/Ada) automates the key activities associated with unit and integration testing. Normally these activities are time-consuming, expensive, and for these reasons often short-changed.

Automatically generates an executable test harness on the basis of one or more files of source code

As illustrated in Figure 2, the executable harness generated by VectorCAST/C++ (VectorCAST/Ada) consists of a test driver, the source file(s) under test, any user-designated stubs for dependent functions, and all un-stubbed dependents. The test harness is data-driven, meaning test data is read by the harness during execution. This approach precludes any need of having to compile and link a new executable harness each time the same tests are performed with different test cases.

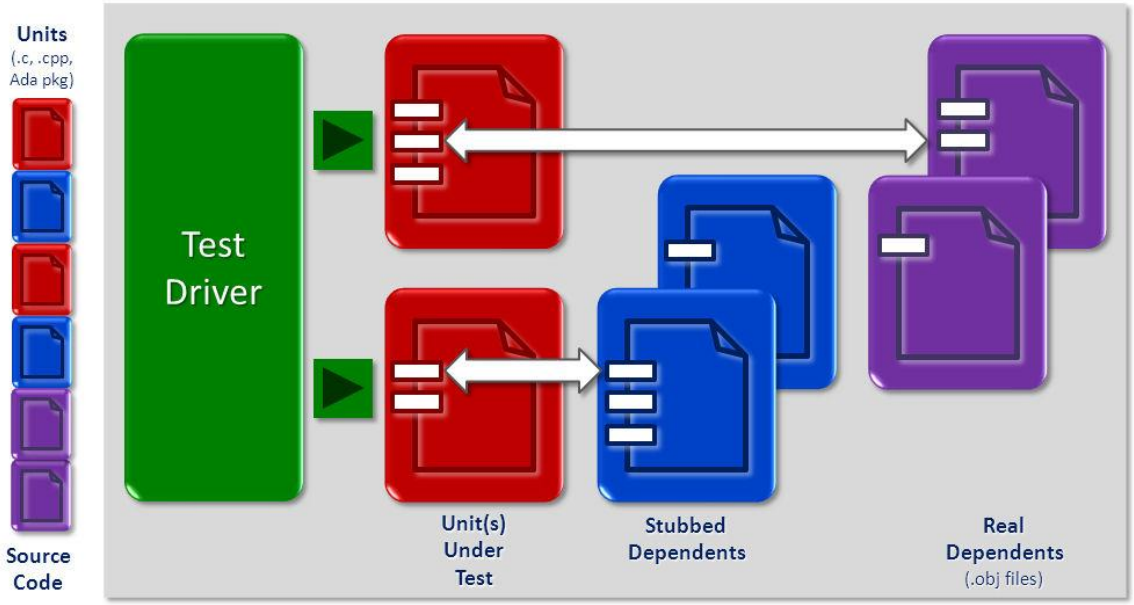

Figure 2: Automated generation of executable test harness

Builds test cases for stimulating the source code under test

Both VectorCAST/C++ (VectorCAST/Ada) can generate test cases automatically, or you can specify the input data and expected results for a specific test. The test cases used (input data and expected results) are automatically preserved for subsequent regression testing.

Executing Tests

You can use VectorCAST/C++ (VectorCAST/Ada) to execute tests in several different ways: (1) on a host machine running natively on a PC or UNIX box; or (2) in conjunction with VectorCAST/RSP, to execute with a simulator or emulator, or directly on an embedded target. Tests can be executed from either the GUI or a command-line-interface (CLI).

Reports pass/fail results and code coverage

VectorCAST/C++ (VectorCAST/Ada) generates pass/fail reports on the tests executed, as well as reports showing the associated code coverage. VectorCAST/C++ (VectorCAST/Ada) can provide coverage information at the statement, branch, and MC/DC levels, as well as at levels A, B, and C for DO-178B.

VectorCAST/C++ (VectorCAST/Ada) can share coverage data with VectorCAST/Cover, and vice versa, to ensure 100% coverage with a combination of system and unit / integration test.

Facilitates regression testing

VectorCAST/C++ (VectorCAST/Ada) maintains all the information it needs for automated regression testing over the course of a developing project. There is no need to manually change or redo test code, as is necessary with many other test framework tools.

VectorCAST/RSP

VectorCAST/RSP allows you to use VectorCAST/C++ (VectorCAST/Ada) to conduct unit and integration tests on real-time applications either on the target itself or in a simulator environment.

VectorCAST/RSP integrates with the target CLI to invoke the cross compiler and to establish an I/O conduit between the target environment and VectorCAST/C++ (VectorCAST/Ada). Test execution is transparent to the user.

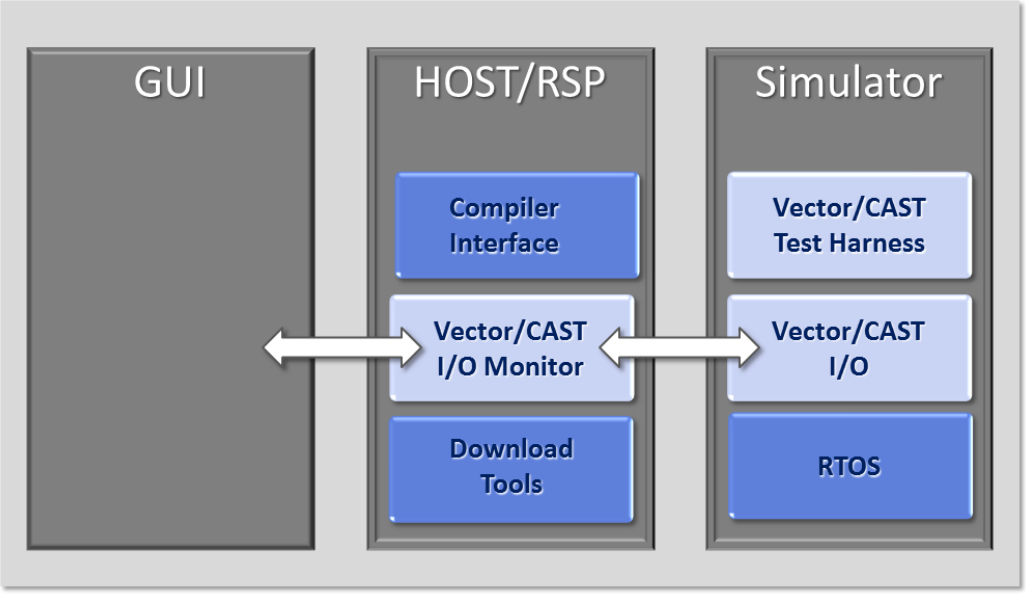

Figure 3: VectorCAST/RSP architecture

In reference to Figure 3, you would use VectorCAST/C++ (VectorCAST/Ada) on a host platform (with RSP enabled) to build a test harness for testing one or more units of application code. The RSP would ensure proper compilation, and would automatically download the executable harness into the target environment.

You would then use VectorCAST/C++ (VectorCAST/Ada) on the host platform to run test cases against the application modules residing on the target or within a simulator. You would also use VectorCAST/C++ (VectorCAST/Ada) to report the coverage of your tests.

VectorCAST/Cover

VectorCAST/Cover allows you to gauge the effectiveness of system testing by determining which areas of an application have been exercised (or covered). You can perform coverage analysis on a portion of an application or on the entire application.

The coverage information generated by VectorCAST/Cover differs from the information generated by VectorCAST/C++ (VectorCAST/Ada). The coverage information generated by VectorCAST/Cover derives from system or functional testing. These tests are usually application specific and constructed to test high-level requirements.

The coverage information generated by VectorCAST/C++ (VectorCAST/Ada) derives from unit or integration testing. These tests are typically constructed to determine whether algorithms are behaving as expected, whether anomalous conditions are being tested, and whether all paths are being tested.

Usually it is not possible to test an entire application during system testing. For this reason, a combination of unit, integration, and system test is necessary to achieve 100% coverage.

Ensuring 100% Coverage

Figures 4 and 5 illustrate two methods of using VectorCAST/Cover and VectorCAST/C++ (VectorCAST/Ada) to ensure 100% code coverage.

Method 1 (see Figure 4) typically applies to the traditional approach to testing, in which the testing progression is from individual components to aggregates, to the full system.

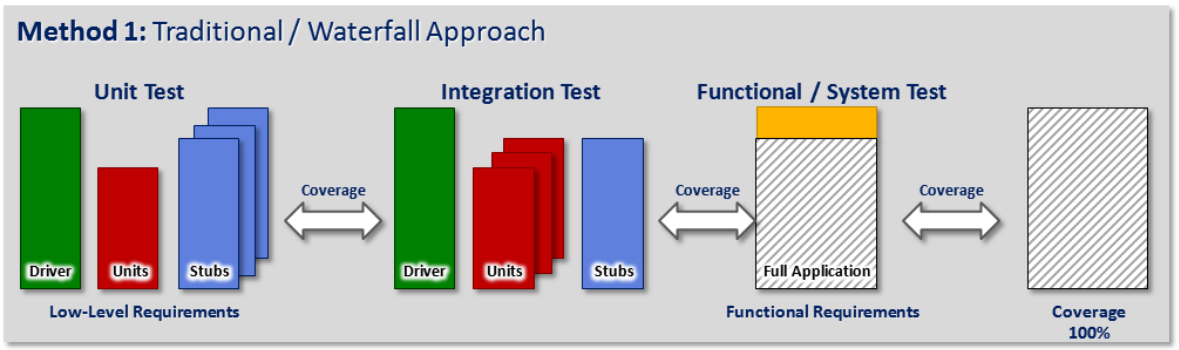

Figure 4: Method 1 – Traditional / Waterfall Approach

Method 2 (see Figure 5) typically applies to requirements-based testing, in which high-level system requirements are typically tested first, followed by low-level unit or integration tests. Achieving 100% coverage is the goal of both methods.

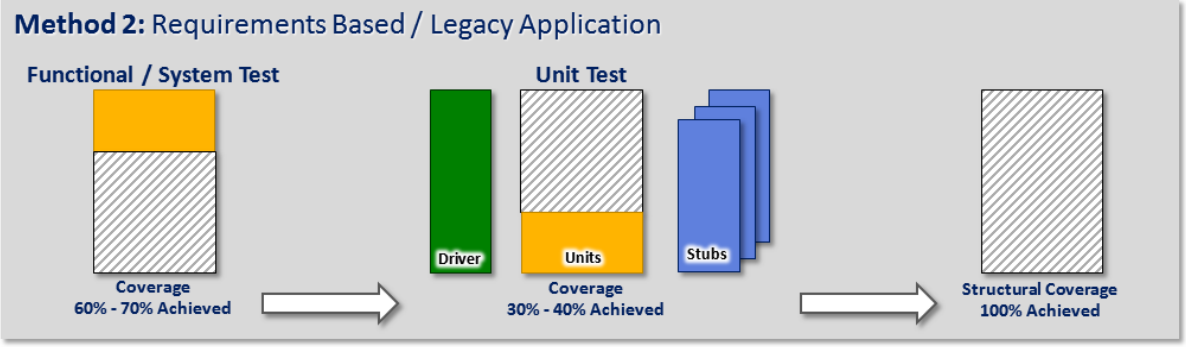

Figure 5: Method 2 – Requirements Based / Legacy Application

Under the first method, you would use VectorCAST/C++ (VectorCAST/Ada) to conduct unit and integration testing on components and aggregates. In conjunction with this testing, you would use the built-in coverage utility to determine and report on source coverage. At functional/system test, you would use the VectorCAST/Cover tool; along with coverage data from VectorCAST/C++ (VectorCAST/Ada) to ensure 100% coverage.

Under the second method, in conjunction with performing system testing, you would use VectorCAST/Cover to identify any 'holes' in source coverage. You would then use VectorCAST/C++ (VectorCAST/Ada) to augment the coverage by running test cases on the files of source code containing 'holes'. In addition, you would use VectorCAST/C++ (VectorCAST/Ada) coverage-analysis utility; along with coverage data from VectorCAST/Cover to ensure 100% coverage.

Note: *VectorCAST/Cover supports the McCabe Cyclomatic Complexity Metric.*

About Vector Software
Vector Software, Inc., is the leading independent provider of automated software testing tools for developers of safety critical embedded applications. Vector Software's VectorCAST line of products, automate and manage the complex tasks associated with unit, integration, and system level testing. VectorCAST products support the C, C++, and Ada programming languages.

Vector Software, Inc.
1351 South County Trail, Suite 310
East Greenwich, RI 02818 USA
T: 401 398 7185
F: 401 398 7186
E: info@vectorcast.com

Vector Software
33 Glasshouse Street, Suite 3.08
London W1B 5DG, UK
T: +44 203 178 6149
F: +44 20 7022 1651
E: info@vectorcast.com

Vector Software
Vorster Straße 80
47906 Kempen Germany
T: +49 2152 8088808
F: +49 2152 8088888
E: info@vectorcast.com