

Leveraging Reusable Tests: Cognos' Long History of Success with Rational Visual Test

When the test engineers at Cognos began evaluating automated testing software, they were looking for a tool that would allow them to use their programming experience to their best advantage.

Having grown tired of the tedium of manual testing and the maintenance burden of home-grown test scripts, the test engineers chose MS Test 1.0, the forefather of Rational Visual Test. In 1992, Cognos created a test environment in MS Test for automated testing of their Powerhouse Client for Windows. Now, six years later, after several product generations and across five distinct product lines, that original environment is still at the heart of ongoing test projects with Rational Visual Test. At Cognos, generating reusable code is not just a design goal for application developers; it is accomplished with great success by the test department as well.

Founded in 1969, Cognos employs more than 1,400 people worldwide with corporate headquarters in Ontario, Canada. Cognos is the leading strategic supplier of enterprise business intelligence tools, allowing users to easily extract critical information through historically separate functions such as data access, reporting, analysis and forecasting. Because Cognos' customers rely heavily on Cognos, Cognos relies on its product testers to assure that all of its award-winning business intelligence tools are of consistently high quality.

Evaluating The Options

Cognos' evaluation process for selecting an automated testing tool centered on three criteria. First and foremost they wanted a flexible tool with a powerful programming language. Packages that focused on, or were limited to, record/playback testing were out. Second, they wanted a tool that would allow them to call into both the Windows API and their own DLLs. Lastly, they needed a tool that allowed them to distribute completed test cases to multiple systems without requiring a separate license for each system.

Dave Moore, Manager of Product Testing for PowerPlay at Cognos, was part of the evaluation team that selected MS Test. MS Test met all the criteria, and while it offered record/playback capabilities, it provided the engineers with unparalleled control via its powerful programming language. MS Test was originally a Microsoft tool that was first sold to end-users in 1992; in the fall of 1996 Rational purchased it from Microsoft. Cognos has been using Visual Test for several generations — from the original MS Test 1.0 to today's Rational Visual Test 4.0. For simplicity, Rational Visual Test and all of its MS Test forebears

W Cognos has been

using Visual Test for

several generations

— from the original

MS Test 1.0 to today's

Rational Visual Test

Time and time

again the evalua-

tions kept coming

back saying the

biggest bang for

the buck was

Visual Test.

are commonly referred to as Visual Test. Moore and his team evaluated and rejected several competing products including an early version of Mercury Interactive WinRunner. WinRunner was eliminated because Visual Test offered much more and it was less expensive -- due partly to the fact that compiled tests could be distributed freely. Cognos had planned from the start to enable its developers to run compiled test cases for themselves before they submitted their code for more thorough testing. Requiring a license for each developer was simply not cost-effective.

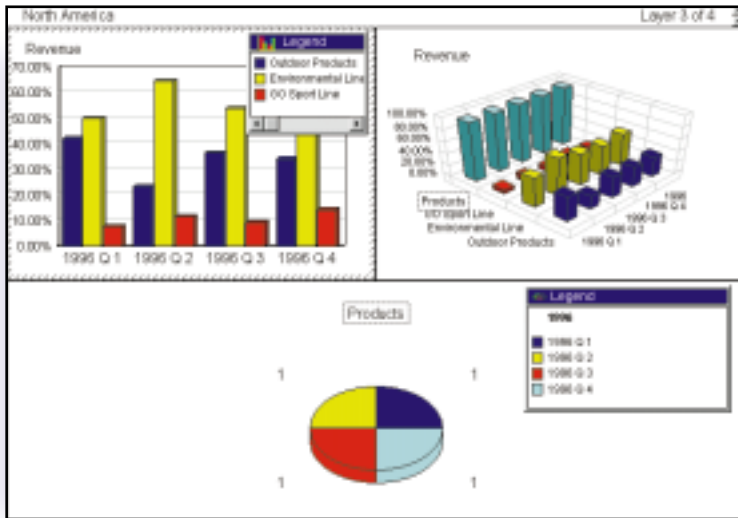
Over the years, Cognos periodically re-evaluated the testing tools on the market, but never came close to switching. "Time and time again the evaluations kept coming back saying the biggest bang for the buck was Visual Test," said Moore.

Getting Started

The automated testing of Cognos' Powerhouse Client for Windows, a 16-bit client/server application, served as the pilot project for Visual Test. The two programmers on the project, Moore and Mike Gabriel, started by creating a few automated test cases. As their knowledge of Visual Test developed, the number of tests they generated expanded quickly. Eventually Moore and Gabriel turned out over 800 automated tests during the year-long project. The time allotted for the project gave them plenty of room to learn and to implement an infrastructure that would be effective and valuable for years to come. Early on in the process a major change to the Powerhouse Client for Windows user interface rendered many of the test cases they had created useless. This led to a test-plan decision that Moore considers most important, specifically the implementation of "wrapper routines" —

a layer of abstraction that enabled test cases to be written in a very general pseudo-code. Details about the actual user interface elements being tested, and details of the test language itself were buried inside these wrapper routines. A change in the application's user interface did not require any rewrite of the main test case, just minor changes to the wrapper routine. The rewards of taking the time to implement this architecture were reaped time and time again as tests were reused across multiple versions of the same product and across product lines.

Much of the verification process for Powerhouse Client for Windows was based on comparing screen images that were known to be correct with screen images generated during the test run, or on comparing large text files that had similarly been generated. While this method is valid, and relatively easy to get working, Moore disliked the maintenance burden it imposed. Approved changes to the application being tested often required that the set of "masters" — the good screen images and text files that are used for later compares — had to be regenerated. This was a time-consuming process, and the sheer size and number of the screen images were causing excessive storage requirements. Also, once a screen image had been generated, it was not necessarily portable across machines; if a particular machine had a different video configuration or color configuration, it could not be used. To alleviate these problems, the team began to tinker with a key feature of Visual Test: its ability to make calls into a Windows DLL. Just as the small time investment in setting up a generalized architecture would pay off, so would taking the time to use this powerful feature.



An example of a Cognos application tested with Visual Test.

Moving Along

As the Powerhouse Client for Windows project was winding down, Cognos began work on Axiant, a much larger effort. During this project, which spanned two years, the test architecture that the team had built was itself put to the test. Moore was pleased with the results, "We took a look at all the test cases we had created for Powerhouse Client for Windows, and realized we could move them to the Axiant environment *without doing any update to them whatsoever.*"

Both Powerhouse Client for Windows and Axiant required verification of many forms, graphs and diagrams that the applications displayed. Realizing that screen capture and compare was inefficient, Moore and his team decided to take a different approach. As the Axiant project was just getting started, the testing team met with the product application development team often, in order to establish a new method for verifying that Axiant was correctly drawing its images on the screen. The development team added test "hooks" to the application's DLLs that could be called directly from the Visual Test programs. These hooks gave the test programs access to

the same memory space that the application was using. "In plain English," Moore explained, "the test program would ask the product — 'You drew the object so now tell me where it is.' " This technique eliminated the vast majority, over 98%, of all screen compares that were being done, and all the hassles that went with them. The test cases actually decided for themselves whether or not they had passed or not. The benefits of being able to call into the DLL did not stop there. It was clear that the product could be tested much more thoroughly: "Without the test DLL our coverage from automated testing would have been around 30%. With the DLL we were able to get into the product in excess of 80%."

Planning and coordination between the product developers and test engineers saved a tremendous amount of time and effort on another key aspect of the test plan. Although the wrapper routines had made the main test cases all but immune to changes in the applications user interface, the wrapper routines themselves were not. For example, a wrapper routine that would verify the label of particular buttons on a dialog box would fail if the labels had been changed for good reason by the application developers. These kinds of changes would have required hundreds of manual modifications to the wrapper routines. To eliminate this maintenance headache, the product application developers agreed to place all of the "define" statements for the application's user interface in a single source file. The test engineers then wrote a script that converted this C++ header file into a Visual Test include file. The wrapper routines would now make use of the resource IDs as defined by the product rather than making direct references to the control through labels. If a change was made to the user interface, the test team would simply rerun the script to regenerate the proper include file. The Visual

Test programs were then recompiled with this newly generated include file and the test engineers were off and running again without having to modify any of the test cases or wrapper routines.

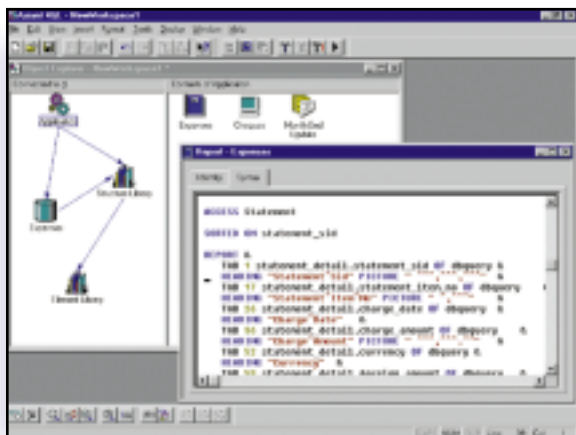
Next Verse, Same As The First

The next Cognos product to be tested with Visual Test was RealObjects. RealObjects is an extension of Axiant, but is strictly a 32-bit application, whereas Axiant is both 16-bit and 32-bit. By the time the RealObjects project started, there were almost 3000 test cases being used to test Axiant. "When we got into RealObjects we took 75% of those Axiant tests and moved them into the RealObjects project — so again we hit the ground running," Moore recalled.

From the very start of the development cycle of RealObjects, the test team implemented standard acceptance tests that were part of the development team's routine. Application developers would check out the source code, modify it to add a new feature, run specific acceptance tests, and if they passed they would check the source code back in. After the developers had successfully verified the new feature with the acceptance tests, the test engineers would run a more complete test suite against the whole product. This approach dramati-

cally reduced the feedback loop to the developers, who knew immediately whether the new feature they had created was working, and whether anything else had been broken. "As a result product stability was way up in a very short period of time," Moore noted. This boost in efficiency did not cost Cognos anything because Visual Test programs can be compiled and distributed to the developers without having to buy a separate license for each of them.

Not only could the developers find out if there was a problem with the code they had just written, they could find out *where* the problem was. The developers were using Microsoft Developer Studio to build C++ applications, and Visual Test is a tightly integrated component of Developer Studio. From the same Developer Studio environment, developers would create and compile the application and immediately run an acceptance test against it. If the application did not pass the test, the developer examined the test case step by step until the problem was detected. Because Visual Test and Visual C++ were integrated into Developer Studio, the developer could then step directly from the test program into the application's C++ source code, at the exact point in the program that the problem had arisen. This eliminated enormous amounts of wasted effort in setting breakpoints to try and pin down the problem. Since Cognos uses Microsoft Visual SourceSafe, another Developer Studio component, as their version control system virtually every aspect of the development routine is tied into one environment. Using a testing tool that was part of this environment was just one more unexpected bonus of choosing Visual Test at the start.



An example of a Cognos application tested with Visual Test.

Given the long

history of success

at Cognos, and the

impressive return

on investment

achieved over

the past six years,

Moore is under-

standably enthusi-

astic about Visual

Test — "It's robust.

It's dynamic.

It's versatile."

Another powerful technique employed in the RealObjects project was the use of data driven testing. The Visual Test programming language includes functions that read and write both text and binary files. The test engineers made use of this capability and wrote tests that opened and read comma-separated value (.CSV) files generated by Excel, and used the data read from the files to drive the test case and verify results. "Data driven tests were very useful to us," Moore commented, "they worked quite well."

The automated testing effort for RealObjects eventually expanded to include 6500 test cases, and was more extensive than either Powerhouse Client for Windows or Axiant. Remarkably, the infrastructure developed for the first project was still in place. "Even today," Moore boasted, "the same environment that was originally created for Powerhouse Client for Windows was continuously updated and modified to turn everything into one big success story."

It Keeps Going, And Going...

After RealObjects there were two more projects within Cognos that required an automation effort, both are still active today. One is Impromptu, Cognos' award-winning data query and reporting tool for the enterprise. Although Impromptu is quite a bit different from the client/server applications which had been part of the Visual Test effort up to this point, the test engineers were confident that they would be able to continue to leverage the work they had been doing for years. "Again, we picked up the whole infrastructure that was used to test RealObjects, which was about 30,000 lines of code, and just tweaked it a wee bit for Impromptu." Moore explained.

The other project currently underway is the automated testing of PowerPlay, Cognos' acclaimed Online Analytical Processing (OLAP) tool. Once more the initial testing environment was set up very quickly due to the same core of reusable code and to the ever-growing Visual Test knowledge base at Cognos.

Moore is leading a team of twenty testers on this project. Everyone in the group uses Visual Test, but only about a third of them are experienced programmers, a third have some programming experience, and the final third have little or no programming experience. Yet another benefit of the wrapper routine architecture implemented long ago is that not every team member is required to be a programmer. Once the team members agree on what wrapper routines are required, what they will be named and what they will do, the experienced programmers can immediately start coding the lower level wrapper routines. At the same time, the less experienced testers can tackle the task of analyzing the application to be tested, determining what is to be tested and how, and then building the high level test cases using a series of calls to the wrapper routines. No knowledge of the Visual Test programming language is required for this group of testers. "Its parallel development," said Moore, "once one part is done, the other is finished." The team members that have little programming experience gain it quickly though. "People don't need to know how to use the functions specifically provided by Visual Test to be productive. Once they become active in creating and debugging the high level test cases, they learn the basics of Visual Test very quickly, as if by osmosis."

Rational Software Corporation

Corporate Headquarters
18880 Homestead Road
Cupertino, CA 95014

Toll-free: 800-728-1212
Tel: 408-863-9900
Fax: 408-863-4120
E-mail: info@rational.com
Web: www.rational.com

Australia
+61-2-9419-0100

Belgium
+32-16-46-24-11

Brazil
+55-11-829-7585

Canada
613-599-8581

France
+33-1-30-12-09-50

Germany
+49-89-628-38-0

India
+91-80-553-8082/9864

Japan
+81-3-5423-3611

Korea
+82-2-556-9420

New Zealand
+64-4-568-9591

The Netherlands
+31-23-569-4300

South Africa
+27-12-663-5677

Sweden
+46-8-703-4530

Switzerland
+41-1-445-36-00

Taiwan
+886-2-720-1938

UK
+44-1344-462-500

...And Going

At some companies the testing efforts for version 1.0 of a product may turn to scrap once development of version 2.0 starts. At Cognos, not only are the tests reused from one version to the next, they are reused on entirely different product lines. Given the long history of success at Cognos, and the impressive return on investment achieved over the past six years, Moore is understandably enthusiastic about Visual Test. "It's robust. It's dynamic. It's versatile."

In the future, Moore expects to continue leveraging all the work that has been put in to date. "Our test routines are moving seamlessly across product lines." Moore continued, "Automation is going full bore here.

There's no looking back. We do keep up to speed on what is new in the product test tool market, however we are not evaluating any other automated test tool at this time. We are completely satisfied with Visual Test."

"With a clear vision, our team of test engineers has made the most of all that Visual Test offers and has transformed our manual testing effort into one that is almost fully automated. We have a dedicated team that extends all the way up to senior managers, who have realized the benefits of good automation, and the fact that it is a software effort all of its own."

Cognos and their customers have been reaping the rewards ever since.