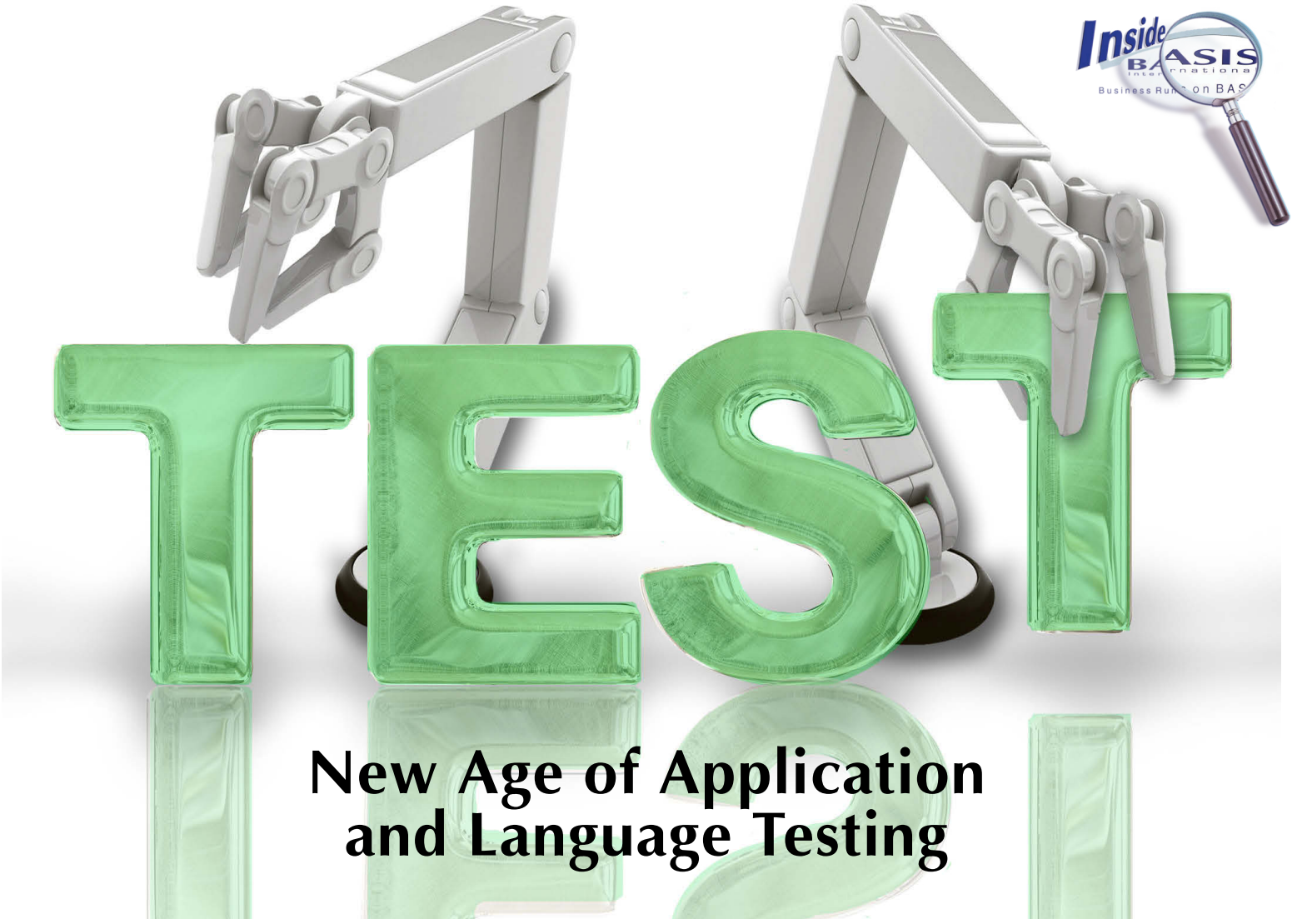


New Age of Application and Language Testing

The process of creating automated GUI tests in any testing framework usually involves a significant amount of time. To BASIS, this investment of time was well worth the effort as we recently created a new GUI testing strategy for AddonSoftware®. This article looks at different testing processes and specifically, the new paradigm BASIS developed to deliver a higher quality product suite.

Testing...a Look at Both Sides

Manually testing any application from start to finish can take days or weeks depending on the complexity of the application in question. This results in time spent not developing new tests and time not testing new features or functionality. Also, running manual GUI tests does not return feedback in a timely manner to the application developers for them to know how their recent changes affected the overall application. Manual testing is also very prone to easily-missed software issues, human errors, and testing mistakes.

By contrast, automated GUI testing tools such as QFTest can run those same repetitive and laborious GUI tests in a fraction of the time it would take a manual tester to perform the same actions. Automated GUI testing gives the application developers feedback on changes in minutes and hours rather than days and weeks.

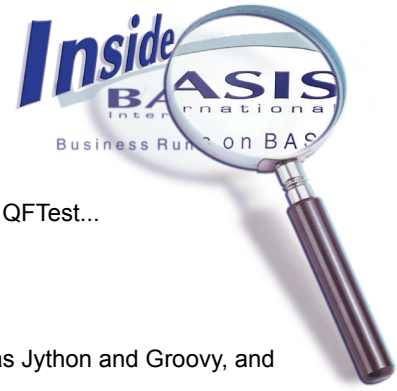
Correctly structured and written automated tests are repeatable, dependable, and reliable. By removing the human factor, automated tests remove the mistakes manual testing is prone to make and catch the errors that humans are prone to miss. Automated testing also encourages developers to try new concepts and add additional innovative features within the product. The rapidity and repeatability of automated testing gives reassurance to the software developers that new code is not affecting other vital aspects of the project.

BASIS...a Look at Their Side Now

To make automated GUI testing possible, BASIS incorporated a number of systems to allow the tests to run nightly in an automated and unsupervised fashion. At the core of the testing model is QFTest, a GUI test tool for Java and the Web. Along with QFTest is an array of other systems – Jenkins, an open-source continuous integration server; the Jenkins Ant plug-in that adds Apache Ant support to Jenkins; and Subversion (SVN), an open source version control system. In addition, BASIS uses the Amazon Elastic Compute Cloud (EC2), a service that provides scalable Windows and UNIX boxes in the cloud. These boxes, or "instances," install and run

By Robert Del Prete
Quality Assurance Engineer

BBj®, AddonSoftware, and the QFTest continually to provide accurate and detailed automated test results of each software build. The BASIS Quality Assurance team then reviews and analyzes these results to help improve the quality of the software and to eliminate bugs and other issues that automated testing uncovers.



BASIS chose QFTest over other testing solutions for GUI-based testing for several key reasons. QFTest...

- is a cross-platform product,
- is capable of testing in GUI, BUI, and Web Start environments,
- has excellent reporting capabilities,
- offers beginners easy record and playback tests,
- provides access to internal program structures through the use of scripting languages such as Jython and Groovy, and
- integrates easily with Jenkins and Ant.

Integrating Various Technologies

BASIS uses various products and technologies to automate the nightly testing of BBj and AddonSoftware. Using Jenkins, a continuous integration server, we can configure a job that automatically launches a new Amazon EC2 instance using Ant and the Amazon EC2 API Tools. **Figure 1** is an example of the Jenkins dashboard that displays all available tasks that run automated QFTest jobs, the time of each job's "Last Success," "Last Failure," and length of "Last Duration."

In **FIgure 1**, the job named "BBj_Windows_QFTest_Testing_Slave_Start" automatically launches a cloud instance that only costs a few cents per hour to run. Once the stave start job completes, the "BBj_QFTest_Testing" job starts, which checks out the latest BBj/AddonSoftware build from the Amazon's S3 Storage bucket onto the cloud instance it just launched.

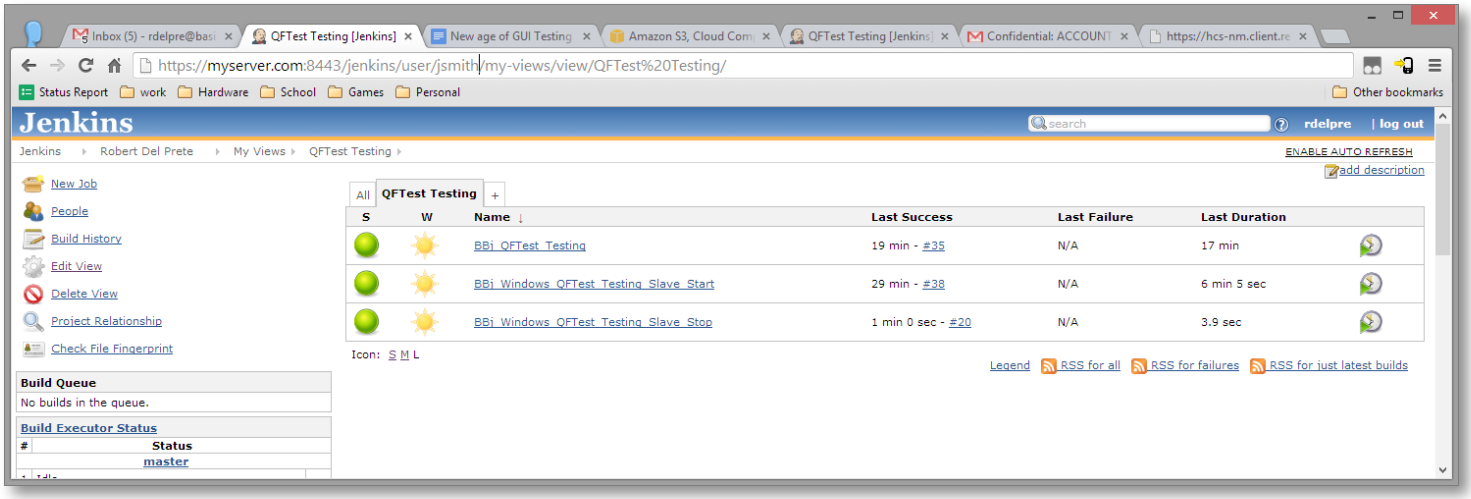


Figure 1. Jenkins dashboard QFTest jobs

Next, this job's Ant script installs the BBj/AddonSoftware build and configures it for testing. The Ant script also checks out all relevant QFTest scripts from the BASIS SVN source code control server and delivers them to the cloud machine. Ant then invokes QFTest to run tests in batch mode and delivers the results back to the Jenkins server for review as artifacts (**Figure 2**) of the job. Once the testing is complete, the Jenkins job "BBj_Windows_QFTest_Testing_Slave_Stop" stops the cloud instance.

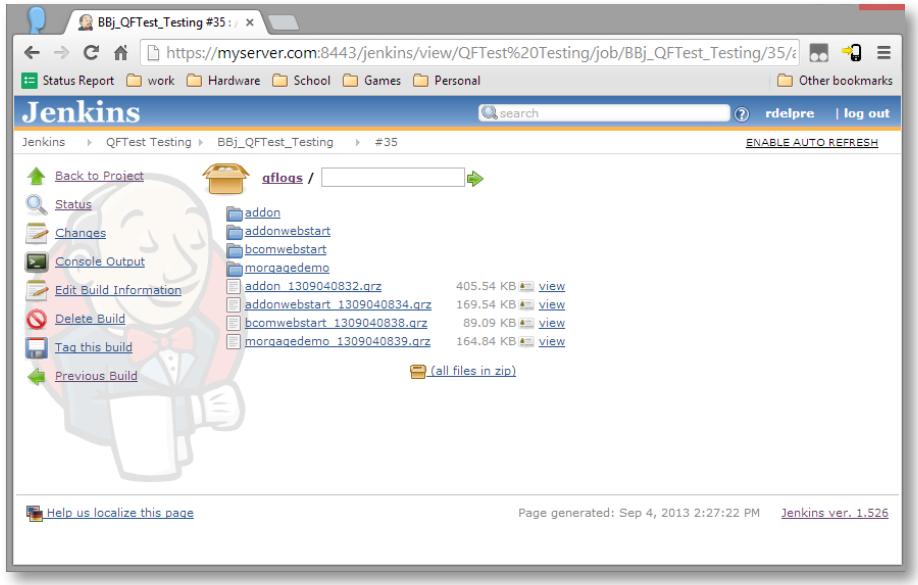


Figure 2. Jenkins artifacts returned after a successful test run

Analyzing Results

One of the most important aspects of automated testing is conveying accurate and timely results of the test runs during the continuous integration cycle. Jenkins can report the success or failure of the QFTest scripts along with the artifacts from the run.

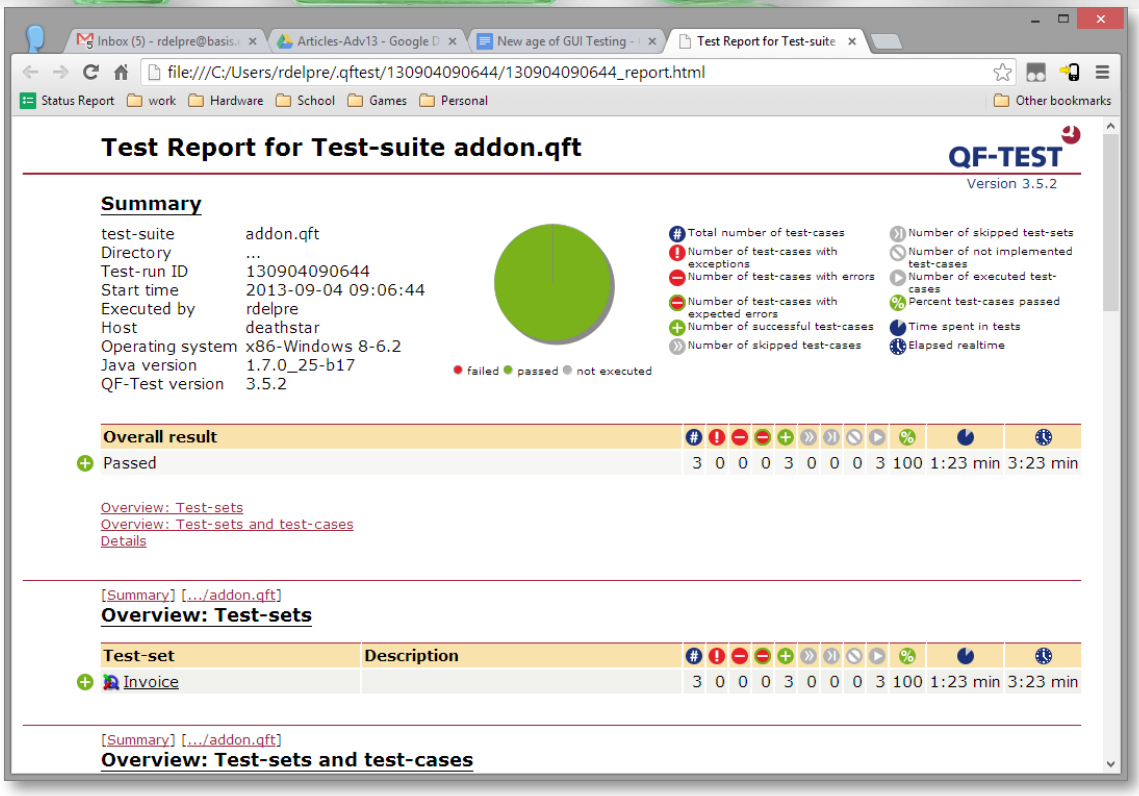


Figure 3. Sample HTML report of an AddonSoftware test run using QFTest

These artifacts include an .html report (shown in **Figure 3**) and a .qrz file, QFTest's version of a log file that is viewable in the QFTest software.

Together, the .html and .qrz log files provide the information needed to fully analyze all the results of a particular test run. These reports reveal any issues or problems with the test while the QFTest log will help to identify specific failures in the "software under test." The BASIS QA team then sends these problems immediately to software engineers for correction.

Summary

Adding automated functional GUI testing into the continuous integration process provides many advantages. Repeatable and reliable tests provide a method to track improvements in product quality and provide engineers with timely feedback on problems that occur with each change to the product line. Automated GUI testing also helps to ensure a better quality product for BASIS developers and their customers. Specifically, QFTest provides BASIS with a way to include GUI, BUI, and Web Start testing into the build process and integrates well with the Amazon cloud and Jenkins. This truly is a new age of GUI testing at BASIS that will never end because there are always more features to test and new ways to test them! ■

For more information on the cloud, Amazon Elastic Compute Cloud (EC2), etc., read
- *Our Salvation is in the Cloud* at links.basis.com/10basiscloud
- *Perfection in the Cloud* at links.basis.com/11cloud
- *BASIS Survived Amazon Outage* at links.basis.com/12survived