TestShell Solutions for Test Automation

Speed business velocity with the industry's easiest to use network test automation capabilities

TestShell offers a fully integrated object-oriented architecture that helps automation projects transcend the limitations of traditional script-based approaches. It allows testing teams to build a library of highly reusable and easy to maintain device resource, provisioning action and testing task objects. A drag-and-drop graphical test creation tool enables non-programmers to easily create sophisticated automated tests and regressions independently. Centralized test execution management, including scheduling and queuing across distributed test execution servers, empowers 24/7 testing.

The Challenge of Traditional Test Automation

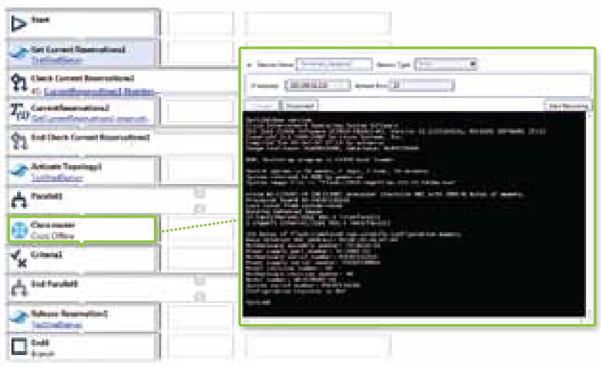
Many test automation projects face challenges due to the limitations of traditional script-based approaches. Script-based automation, whether created natively by programmers in TCL, Perl or Python, or via proprietary record-and-replay automation tools, suffers from significant limitations:

- Costly to maintain or refactor due to script complexity
- Programmer-dependent - preventing the majority of test engineers from effectively using the scripts, and a source of high project risk in the event of programming personnel turn over
- High TCO - in order to adjust to the changing test environment there is a hefty script bloat and a huge waste of programming time on developing minor variations to a relatively small number of scripts
- Low penetration - many automation projects do not exceed 10% penetration due to low re-usability and dependence on programmers
- Poor ROI and project fatigue - failure to accelerate test cycles offers a poor ROI, potentially resulting in project fatigue, loss of funding and collapse of the automation process

TestShell: Object-Oriented Automation

TestShell changes test automation projects by providing an object-oriented framework; every element and process created and managed by TestShell is object-based. Test infrastructure resources such as DUTs and test generators, provisioning actions and test tasks are captured as highly reusable objects, making them easy to share, refactor and maintain as environments change.

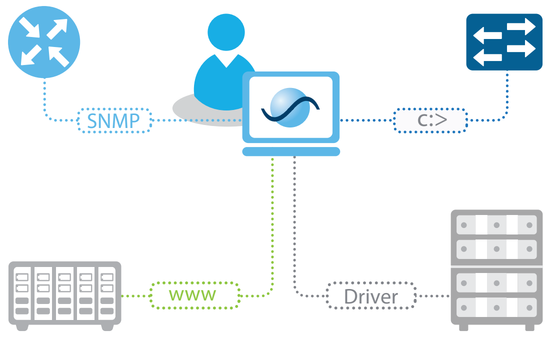
TestShell's object library is leveraged by non-programmer friendly automation design tools to drive high levels of automation penetration into the testing process. Powerful test execution and reporting ensure highly scalable and resilient testing operations. **TestShell's extensive capabilities deliver a compelling return on investment (ROI) and low Total Cost of Ownership (TCO).**



Open Device Control - for quick and effortless integration

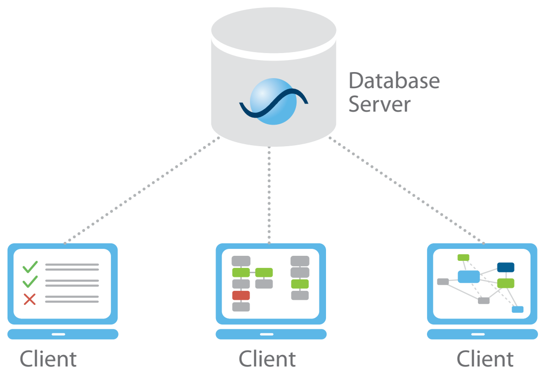
TestShell provides complete control of physical and virtual devices in the testing environment. An open device driver builder feature enables testing teams to easily create device control objects and quickly refactor them to adapt to changing conditions and requirements. TestShell provides a huge variety of interfaces, including:

- Control interfaces (Telnet, SSH, Web Services, Serial)
- GUI with full object capture (Windows, Java, Web)
- Script-based API (TCL, Perl, Python)
- TestShell control libraries of leading brands (Ixia, Spirent, Shenick, MRV, OnPath)
- Vendor and custom drivers (.NET, Exe, ActiveX)
- Management applications (HP QC, IBM RQM)



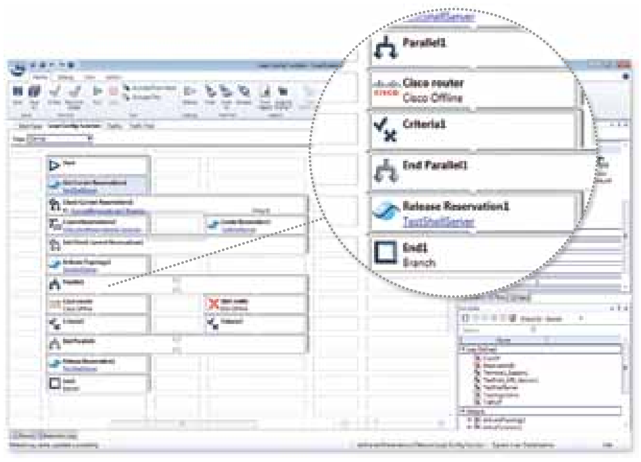
Central Object and Test Library – for increased productivity

TestShell delivers carrier-class scalability through a multi-tier client-server architecture. A centralized server stores and synchronizes all automation objects and workflows. This maximizes team expertise, increases automation reuse, and optimizes knowledge transfer, resulting in much higher penetration of automation into testing processes.



Flowchart-Based Automation Design

TestShell's advanced GUI-based test editor enables engineers to directly create automated test scenarios leveraging a shared object library of test resources, provisioning actions and testing tasks. Through the application, test engineers can drag and drop objects onto a flowchart canvas, add logic and create sophisticated workflows that issue interface commands, analyze response, marshal information between steps, utilize parallel sequencing and generate real-time report data.

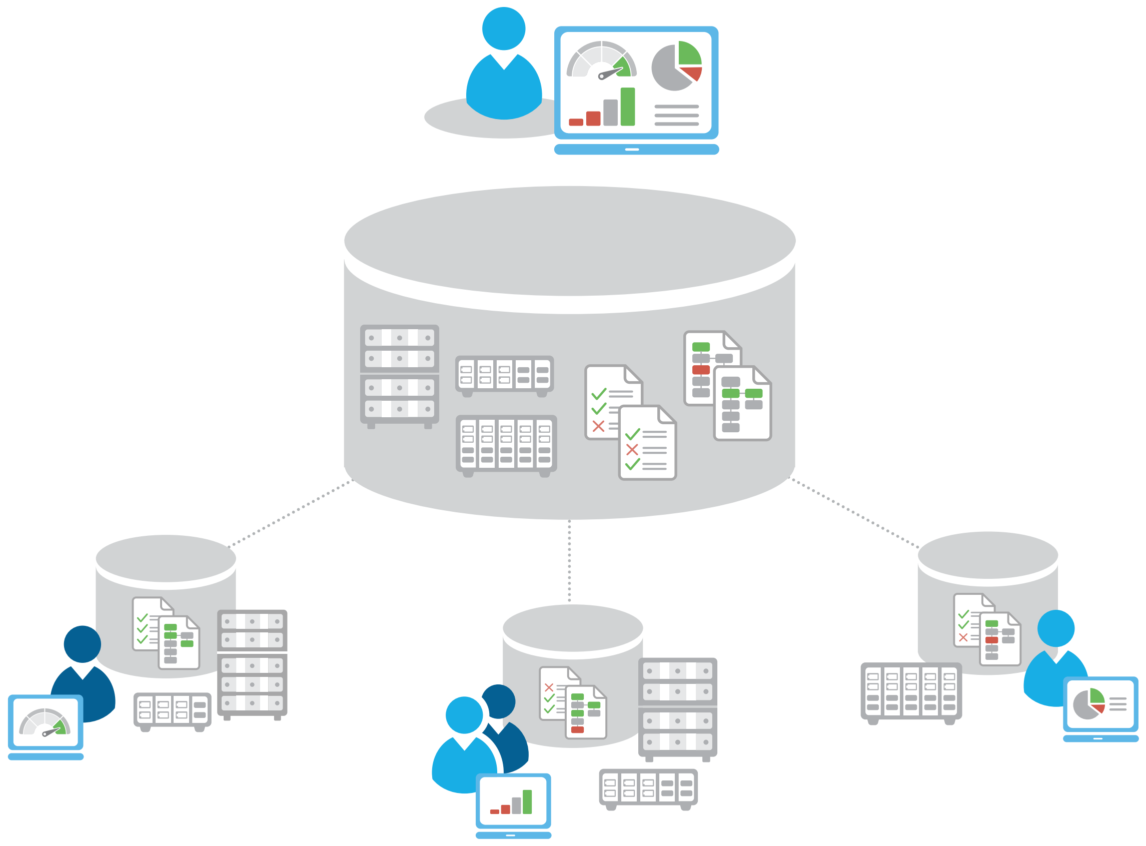


Scalable and Resilient Test Execution - for reduced test run duration

TestShell optimizes test execution by centrally controlling test runs across multiple stations and execution servers. Tests can be launched through a time-based scheduler and are optimized through flexible and resilient queuing. TestShell locates available stations and remotely launches required tests in sequence, maximizing station utilization and minimizing overall test run duration.

TestShell

QualiSystems



Central Results Collection and Aggregation - for fast reporting and analysis

All TestShell test data and results from cross-site stations are automatically stored in a central server, using a unified format. Powerful "database jobs" aggregate and prepare the data for simple and quick querying and reporting. Advanced data tagging mechanisms enable users to associate test runs and measurements with devices and other parameters such as device version, part number, project, test plan, requirements and more. Advanced reports are generated in a click of a button.

Online Customizable Dashboards and Reports - for well informed decision making

TestShell improves management tracking and trend identification by displaying measurements and test results in real-time. A powerful query builder enables users to mine any information from the database. Customizable reports and dashboards are easily created and displayed, offering a snapshot of business-critical information such as test coverage and progress, failure count per version, measurement trends, and release status.

TECHNICAL SYSTEMS INTEGRATORS

To learn more about TestShell, visit our website:

www.tsieda.com
sales@tsieda.com
(407) 339-4874, ext 111