The TestShell software framework helps telecom service providers, data centers, enterprise IT and equipment manufacturers to optimize lab performance and accelerate testing!
The TestShell Framework is an extensive application suite offering complete Lab Management, Device Provisioning and Test Automation solutions for engineers, enabling them to optimize lab performance and increase testing coverage while expanding equipment utilization, reducing setup time and accelerating testing.

**TestShell provides:**

**Advanced Lab Management**
- Central Device Inventory
- Reservation and Scheduling
- Topology Creation
- Embedded Layer 1 Switch Control
- Device Utilization Reporting

**Extensive Device Provisioning**
- Drag-and-Drop Integration Editor
- Live Topology Diagram
- Automated Maintenance Procedures
- Setup Validation and Troubleshooting

**Robust Test Automation**
- Device Control
- Flowchart Based Test Editor
- Central Test Management and Sharing
- Distributed Execution and Test Scheduling
- Central Results Collection and Aggregation
- Online Customizable Dashboards and Reports

**TestShell Value Proposition**
- Optimize lab performance
- Increase testing coverage
- Maintain team independence
- Shorten time to market

TestShell enhances quality and boosts performance, enabling your organization to achieve more in a shorter time-frame, maximizing existing resources.

The framework is suitable for various types of organizations including carriers and service providers, enterprises, data centers, network equipment manufacturers, storage and flash memory manufacturers and others.
**Central Device Inventory**

for increased utilization

Central management of all lab equipment, including powerful device tagging based on parameters like model, type, capabilities and cabling, simplifies device tracking and sharing for increased device utilization.

**Reservation and Scheduling**

for easy sharing

Scheduling and coordinating equipment and topologies through reservations on a shared calendar. Optimize equipment utilization and sharing while eliminating overlapping and test breaks through a powerful conflict resolution mechanism.

**Drag-and-Drop Topology Creation**

for easy setup

Quick and easy creation of complex topologies using a drag-and-drop user interface allowing easy modification and reuse. Quickly find available equipment through powerful search grids, add them to the diagram and graphically define the required connectivity.

**Layer1 Switch Control**

for effortless connectivity

Embedded control of physical switches including MRV, ONPATH, APCON and more is available. Topologies are drawn without the need to define the physical switches. The system will translate the path requests through the hidden switches and automatically configure all of them. Topologies are simplified and connectivity is immediate.
Utilization Reporting - for accurate tracking

Simple generation of lab performance and utilization reports provides organization-wide usage tracking enabling quick identification of bottle necks and lack of usage. These reports support lab managers’ decision-making on future equipment purchase and improve device allocation.
Device Provisioning

Drag-and-Drop Integration Editor for building it yourself
TestShell supports vendor-agnostic control of all network devices through integrated interfaces that communicate with the networking and testing equipment. Through the intuitive drag-and-drop driver editor, direct interactions with devices are presented to the user on the topology diagram. Engineers with no programming skills can now join the automation creation effort.

Live Topology Diagram for efficient user control
TestShell replaces static environment diagrams with a live and active workspace. Accessible and dynamic topology diagrams allow you to constantly update and change your setup to meet your requirements. Equipment control procedures are easily activated with quick and effortless provisioning of the complete topology in a single action. Engineers can now easily set up their testing environment for manual testing, automated testing and other activities.

Automated Maintenance Procedures for increased efficiency
Schedule nightly, daily or weekly configurable maintenance procedures that check the status of devices, update the central repository, rollback devices to past configurations, alert on different user-defined events, and many other options. With TestShell users can easily create their own maintenance procedures and rules and consequently run their test lab efficiently.

Setup Validation and Troubleshooting for increased reliability
Launch topology and device validation procedures that can verify whether the required set up was actually deployed. Execute and launch troubleshooting procedures in order to fix or alert about malfunctioning devices. Increase setup reliability and minimize lengthy debugging of complex environments.

www.qualisystems.com
**Device Control**
for quick and effortless integration

Quick, user-friendly and complete control of required equipment in the testing environment. Easily interact with all the devices sending different commands through a huge variety of interfaces, collect the different responses and continue processing them.

**A few examples of interfacing methods include:**

- **Control interfaces**
  Telnet, SSH, Web Services, Serial and more

- **GUI**
  Windows, Java, Web and more

- **Script-based API**
  TCL, Perl, Python and more

- **TestShell control libraries of leading brands**
  Ixia, Spirent, Shenick, MRV, OnPath and more

- **Vendor and custom drivers**
  .NET, Exe, ActiveX and more

- **Management applications**
  HP QC, IBM RQM and more

**Flowchart-Based Test Automation Scenario Editor**
for fast test creation

TestShell’s advanced GUI-based test editor enables engineers to directly create automated test scenarios. Through the application, flowchart test sequences are quickly created driving the different interface commands while analyzing responses, marshalling information between steps and defining flow rules. The graphical grid canvas allows simple generation of parallel sequencing dramatically reducing test execution time without increasing development complexity.
Central Test Management and Sharing for increased productivity

TestShell is based on a distributed database server that allows automated storage and synchronization of all test scripts and data across sites. Comprehensive test asset deployment and reuse is available for all users, maximizing productivity and knowledge sharing.

Distributed Execution and Scheduling for reduced test run duration

TestShell optimizes test execution by centrally controlling test runs across multiple stations. Tests can be launched either through a time-based scheduler or through a queue of tests in a test plan. The central control will locate available stations and remotely launch the required tests, maximizing station utilization and minimizing overall test run duration.

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 & 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 like: Test coverage and progress, failure count per version, measurement trends, release status and more.
**TestShell Applications**

- **TestShell Resource Manager**
  An application for managing and reserving lab equipment combined with setting up topologies for accelerated lab performance and equipment utilization.

- **TestShell Driver Builder**
  Device interface editor for fast driver creation enabling engineers to independently and easily build equipment control packages.

- **TestShell Studio**
  A graphical, easy-to-use, test authoring application enabling quick development of intricate test scenarios using a drag-and-drop interface.

- **TestShell Runner**
  A family of fast, fully-automated test and provisioning executors that provides uninterrupted operations on any device or system to ensure maximum productivity without any data loss.

- **TestShell Controller**
  A central test execution manager enabling both calendar based planning of test cycles and creation of large test cycle queues to be executed on multiple TestShell Runner stations.

- **TestShell InSight**
  An online web application providing advanced user-defined dashboards and reports in real-time.

- **TestShell Server**
  A distributed database server that provides the vital infrastructure for each component of the TestShell suite, and allows them to work seamlessly together. The server is available in different configurations to meet specific customer requirements.

[www.qualisystems.com](http://www.qualisystems.com)
TestShell is available in three editions, suited for large enterprises as well as medium and small organizations:

**Enterprise Edition**

**Large scale Test Automation across sites**

TestShell Enterprise Edition is a comprehensive test automation and lab management framework for large-scale enterprises with multiple, and geographically distributed, testing teams and labs. The fully scalable Enterprise platform offers comprehensive lab and resource management and test execution management across multiple stations. The Enterprise Edition can be further enhanced with additional TestShell modules for advanced business intelligence.

**Team Edition**

**Sharing Test Automation throughout the organization**

TestShell Team Edition is a test automation and lab management framework for medium-size organizations with single-location testing teams and labs. The Team Edition is suitable for organizations requiring up to ten test authoring and execution stations or ten active reservations. TestShell Team Edition offers advanced sharing capabilities for test data and resources among testing teams. The Team Edition can be enhanced with additional TestShell modules for comprehensive lab and resource management, test execution, and advanced business intelligence.

**Personal Edition**

**Complete Standalone Test Automation Station**

TestShell Personal Edition is a basic standalone test automation framework for small organizations. The Personal Edition covers the classic stages of test automation – test authoring, execution and reporting in a single station.
## Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>Personal</th>
<th>Team</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central device inventory</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Device reservation and scheduling</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Drag-and-drop topologies setup</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Embedded Layer 1 switch control</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equipment utilization reporting</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Drag-and-drop device integration editor</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Live topology diagrams for accessible and dynamic setup</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Automated device maintenance procedures</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Setup validation and troubleshooting</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Vendor-agnostic control of all network and test equipment</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Drag-and-drop creation of intricate automated scenarios</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Central test management and sharing</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Integrated parallel execution</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Real-time viewing of results and raw data during test execution</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Distributed execution and scheduling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automated test scenario documentation &amp; storage</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Simultaneous test execution across multiple stations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Central results collection from multiple sources</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Automated customized report generation</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Online customizable dashboards and reports</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Over 10 test authoring and execution stations</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Central repository across sites</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>TestShell server configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standalone</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enterprise Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional additional modules</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TestShell server configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standalone</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enterprise Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controller</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Resource Manager</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>InSight</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Driver Builder</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Resource Manager</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>InSight</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Driver Builder</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

### TestShell Value proposition

- Optimize lab performance
- Increase testing coverage
- Maintain team independence
- Shorten time to market

www.qualisystems.com
TestShell

- All inclusive: A single framework reduces bureaucracy and improves collaboration
- Scalable: Vendor agnostic integration to any network and test equipment
- Dynamic: Flexible test environment simplifies provisioning and modification
- Easy to use: Testing team independency saves time and overhead costs

About TestShell

TestShell is an end-to-end software framework for lab management, device provisioning and test automation. The software is used by carriers, network service providers, data centers, enterprises, and device manufacturers. TestShell provides complete lab management, resource scheduling, automated device provisioning, topology creation, test automation and comprehensive reporting and analysis. TestShell helps organizations to improve product quality, optimize lab performance, shorten time-to-market, and significantly reduce capital and operational expenditures.

About QualiSystems

QualiSystems is an automation pioneer providing organizations with unparalleled Test Automation and Lab Management solutions that drive innovation, efficiency and ROI. QualiSystems TestShell Framework has already proven itself as an industry-critical solution in North America, APAC, Europe and the Middle East, where it is used by market leaders from a wide spectrum of industries.

www.qualisystems.com • info@qualisystems.com

© All other company, brand and product names are marks of their respective holders.
QualiSystems Ltd. Patent pending. All rights reserved.