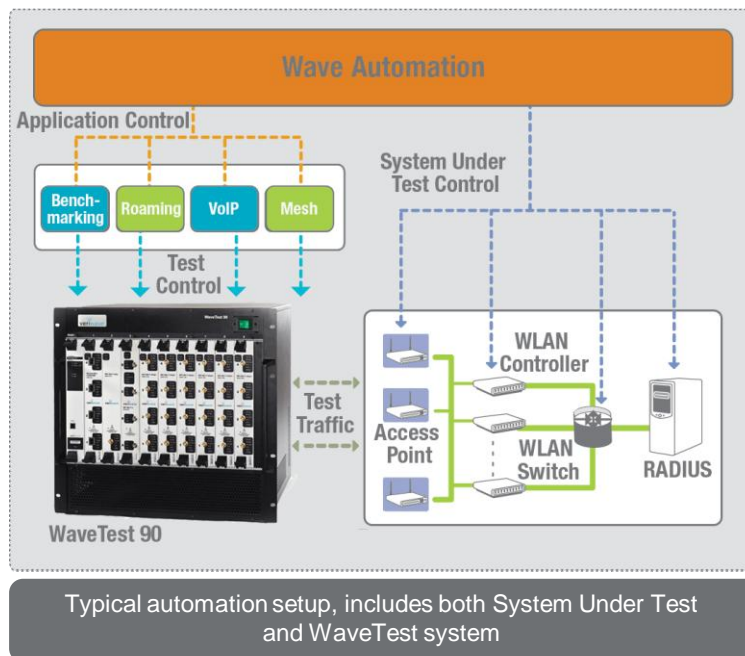


WaveAutomation[™]

WaveAutomation offers a complete customized solution for automating WaveTest platform applications as well as the System Under Test (SUT). It provides an automation framework and services that allows for comprehensive user control and programmability along with the convenience of running fully automated tests while maximizing test coverage and reducing time to market.



Benefits

- Execute multiple test sequences with the ability to change parameters
- Provide extensive test coverage in a short amount of time
- Dramatically reduce test setup, execution time and labor cost
- Full control over WaveTest and SUT parameters used by each test
- Replaces many of the repetitive, time consuming tasks associated with testing new software releases
- Run un-attended tests for days or weeks in a row
- Eliminates manual test results analysis and error checking
- Front-end Tcl script interfaces are easily customizable to support new test beds
- Highly suitable for regression testing
- Complete integration with APs, WLAN controllers, switches and RADIUS servers that eliminates manual device configuration

Features & Utilities

USER SPECIFIED PASS/FAIL CRITERIA

- Allows users to establish the acceptance criteria for each type of test
- Test return status is based on the outcome of the pass/fail criteria. For instance, a PASS return means that the test met or exceeded the user established criteria
- Pass/Fail results included in the PDF reports and csv data files.

DATA EXPORT TO MYSQL

- Allows for exporting test results into an external MySQL database. Exporting to other databases or data collections forms available through VeriWave professional services
- Includes multiple tables for tracking test case execution, per test type results, session tracking, and configuration tracking
- Allows for consolidation of results across many different test automation runs
- Each test type (e.g. latency, throughput, voice capacity, etc.) exports the specific and pertinent data for the test
- Exported data includes the pass/fail status of the test
- Data export API in Python

Methodology

WaveAutomation makes it possible to configure fully automated test suites, which can either stand alone or be integrated into an already existing automation system. Documentation and sample configurations are provided enabling new users to quickly become oriented and productive with WaveAutomation. All aspects of the tests are controlled by WaveAutomation, including:

- Configuring the WaveTest system with test-specific parameters such as client topology, traffic patterns, frame sizes and authentication types
- Configuring the Devices Under Test (DUT) such as APs, WLAN controllers, switches, and RADIUS servers, using native command line interface (CLI)
- Executing multiple tests while monitoring hardware status and results
- Comprehensive error checking and event reporting
- Summarized and detailed reports showing statistics counters, graphs and historical test logs

WaveAutomation provides an easy to use Tcl-based configuration interface that allows the user to specify a sequences of test to execute. In addition, multiple parameters for each test can be specified and configured using the WaveTest system and the hardware components on the SUT such as APs, WLAN controllers, switches or RADIUS servers. The configuration interface files can be shared between users and are easily modifiable.

Configuration & Control

WaveAutomation manipulates the test parameters defined by WaveApps. It provides an alternative Tcl-based interface to configure multiple combinations of parameters and mappings, in addition to configuring the hardware components of the SUT:

	WaveApps	WaveAutomation	SUT
Test execution	Single	Multiple	
Traffic mapping, frame sizes, intended load, PHY rate	√	√	
Radio types (a/b/g/n)	√	√	√
Client identities, pre-shared keys, client server certificates	√	√	
Number of trials and trial duration	√	√	
RADIUS server address			√
IP address / port configurations	√	√	√
Authentication and encryption types	Per Test	For all Tests in List	√
Pass / Fail criteria		√	N/A

Minimum Requirements

Operating System	Linux (2.6 Kernel) or Windows XP SP2
Packages	ActiveState Tcl 8.4, tclx-8.4, expect, ActiveState Python-2.4.2