# TestShell Studio Training

Student Handouts

--------------------------------------------------------------------------------------

Author name : Shay Cesary

QualiSystems

## Table of Contents

## Table of Figures

## Preface

This document is a supplement of the TS-S TestShell Studio Basic Training. The handouts are not replacing the training presentations, but to supplement and assist users.

The basic assumption is that, the user has Studio client or standalone versions already installed on the user's PC or Laptop. This document will not be relevant if a working, TestShell Studio™ is not available.

The user is referred to as "He" for convenience only. This document aims for men and women alike.

## Quick Start

To start the system, please follow the steps below:

1. Click on the Studio Icon

    (Alternative: Start Programs->QualiSystems TestShell->TestShell Studio)

2. After the splash screen finish the load up, type your credentials

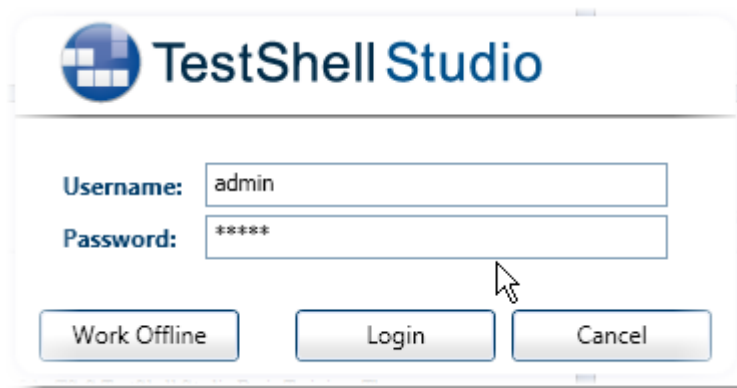    (by default - User: admin Password: admin)



Figure 1 - The Studio Login

3. The Start page appears; select the option New Test from the "Actions" section
    (The middle of the document). Alternatives: Select "New Test" from the top
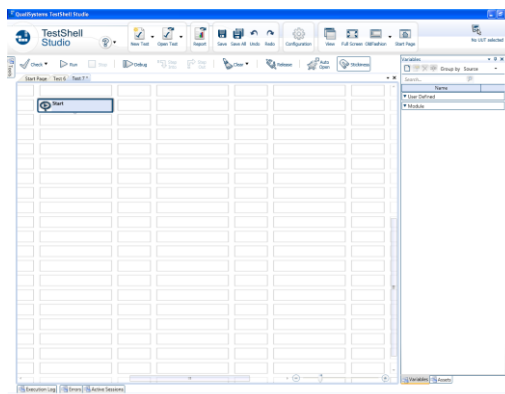    ribbon; Create a new test on the "Tests" pane, double click to open it



Figure 2 - The studio canvas with a new test

## What is testing?

The standard concept of testing is structured from these basic steps:

1. **Setup**: Creating the initial conditions for testing: Connections, inputs, environment, control signals, etc. This stage may include instructing users

2. **Measurement**: Reading from a device the results of the tested action. In some cases, there is a need for a device to transform physical attribute (voltage, temperature, packet rate, loss) to digital information.

3. **Analysis**: In most cases, the needed data is not readily available from the measured data. This stage can take many different forms:

   a. Text parsing: the information is part of a larger text, and the information needs to be extracted according to markers in the text
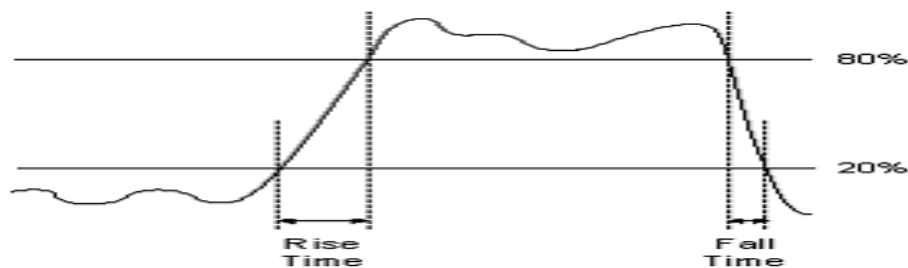
   > Ping statistics for 192.168.42.1:
   >
   > Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   >
   > Approximate round trip times in milli-seconds:
   >
   > Minimum = 0ms, Maximum = 0ms, Average = 12ms

   b. Graph Analysis: the information is analyzed mathematically from the output graph (Rise time, fall time, IP3, spike, spurious, etc.)

   

   c. Conversions (Hex to Decimal, String to array, H/Vlookup, etc.)

4. **Decision**: Once the needed data is processed, the test can reach a decision about the results: Comparing to a 'golden' result, compare to a specification document, white/black lists, search for a specific string, upper/lower limits

5. **Reporting**:

   Display on the report, add text/image/link, add metadata tags for future BI queries

QualiSystems

*Confidential*

| Step | Basic Electronics | RF | Telecom | Medical |
|---|---|---|---|---|
| SETUP | Set a power supply output voltage to 7.5 Voltage. Set environment chamber to 75$^{o}$C | Set the input voltage, a network analyzer input and the power meter range | Set a traffic generator with TCL script. Program a router through serial port | Set a black body radiator, program a pulse and breathing simulator with VB script |
| MEASUREMENT | Measuring the device output voltage from an oscilloscope | Capturing the network analyzer and spectrum analyzer graphs | Read the output log. Parse the relevant information | Reading thermometer, Pulse reader and Oxygen absorption levels |
| ANALYSIS | Calculating Rise Time, Ripple and Output Voltage. | Calculate: Amplification, BW, IP3 and SNR. Detect spurious | Parse from the result report the loss rate, error rate. Detect DoS and load balancing problems | Compare output graphs to golden graphs – calculate correlation |
| DECISION | Rise time<5msec, Ripple<5%, 4.8V<Output voltage<5.2V | 10dB<A<11dB .74Khz<BW<.76Khz IP3>50. Spurious less than 2dBm | No decision – test is continues | If any of signals is missing – Fail If correlation between golden and actual graphs less than 97% - fail |
| REPORTING | Display the scope graph and numeric results. Add the environment conditions | Display Graphs; attach system spec document and numeric results. Add user digital signature | If loss rate is less than requires – send mail to the field office hotline | Display graphs; attach numeric reads of all sensors. Attach the FDA approval note to each report |

## Authoring Tests

TestShell Studio test authoring is simple: Drag, drop and configure.

### 1.1     Scenario

Let us create a simple test – we will check if the room temperature is between $20^{o}$C to $25^{o}$C. We will ask the user to inspect a thermometer to get the temperature, we will check if the recorded temperature is within the range (20 to 25) – if it is above the limit – instruct the user to activate cooler; if it is below – instruct the user to activate heater.

### 1.2     Step-by-step

a) Create a new test (see Quick Start above)

b) From the Tools pane (right hand), select the utilities stencil, and drag the INSTRUCTION module just below the start step (see Figure 3) Double click the instruction step to edit

c) In the main field, type "Please type the temperature in the room (In centigrade)"

d) Change the user output label to "Temp"

e) Change the output type to Numeric

f) Now, to set the limits, click on the "Set Criteria" button
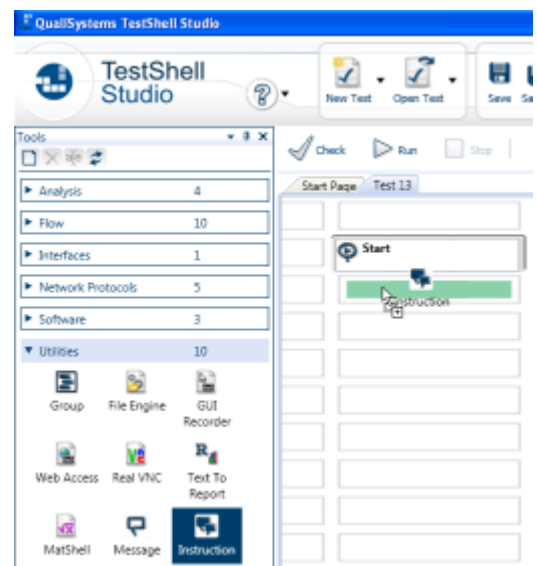


Figure 3 - adding an Instruction step
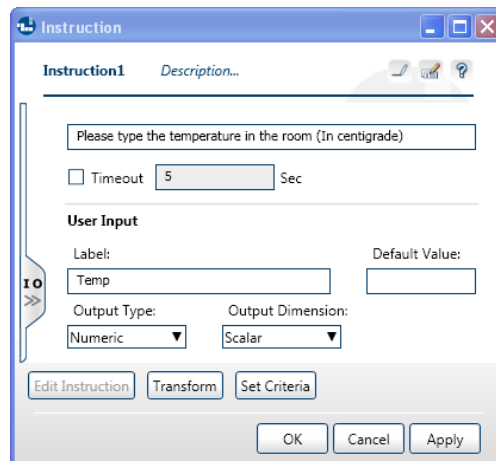
Figure 4 - Editing the instruction

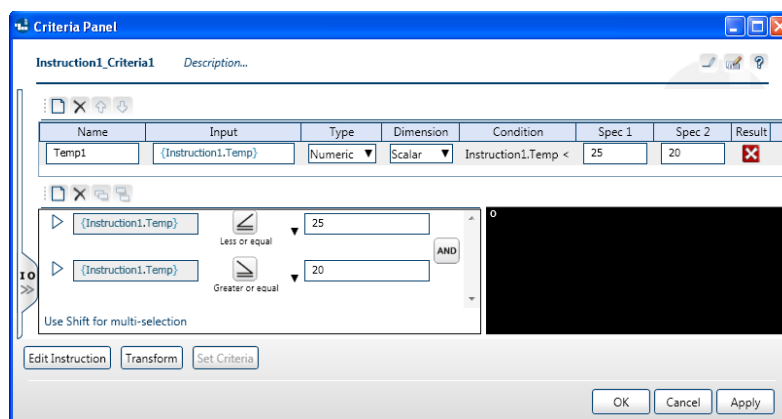g) Set the limits for the temperature (upper=Spec 1, lower=Spec 2)



Figure 5 - setting the criteria

h) Click ok to close the module

i) Now we need to determine if it was too hot, too cold or just right. Drag a CASE module from the Flow stencil, right after the previous module
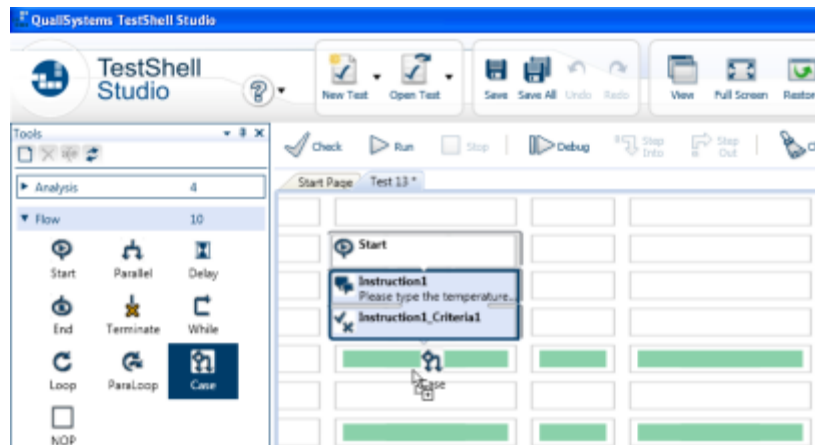
Figure 6 - placing a case

j)  By default, a case has two options. We need three (Hot, Cold and OK). Click on the right edge of the case module, and drag it to add another option

k)  Double click the case module, there are three tabs 1, 2 and default. We need to set the conditions for 1 (too hot) and 2 (too cold)
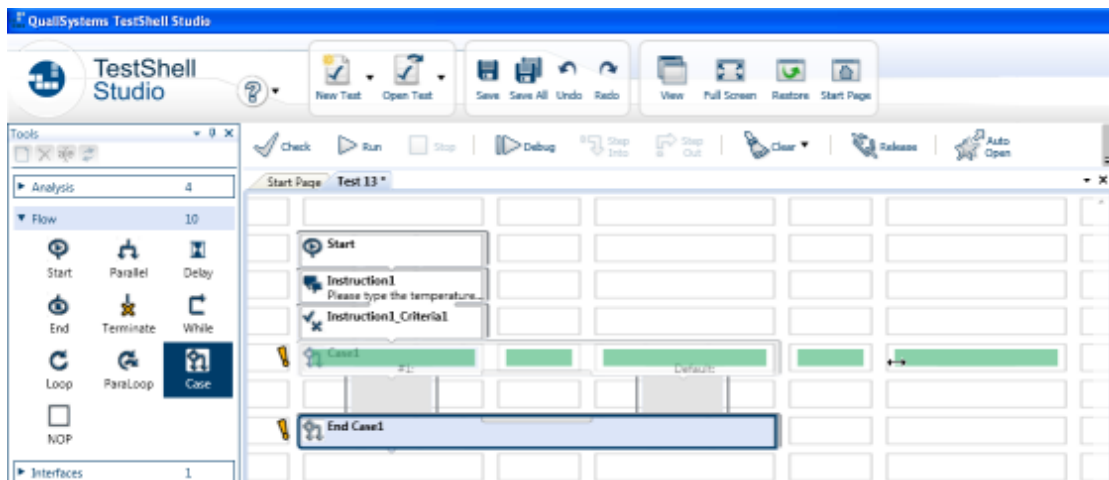


Figure 7 - Adding another path to a case

l)  Select the first condition (tab 1). In the left field, right click and select "insert variable". Insert the output of the Instruction module *INSTRUCTION1* which we labeled *Temp*. All expressions, variables and functions are enclosed in curly brackets. So once selected – the left field will be {Instruction1.Temp}. Mark the color of the test – indicating a valid variable
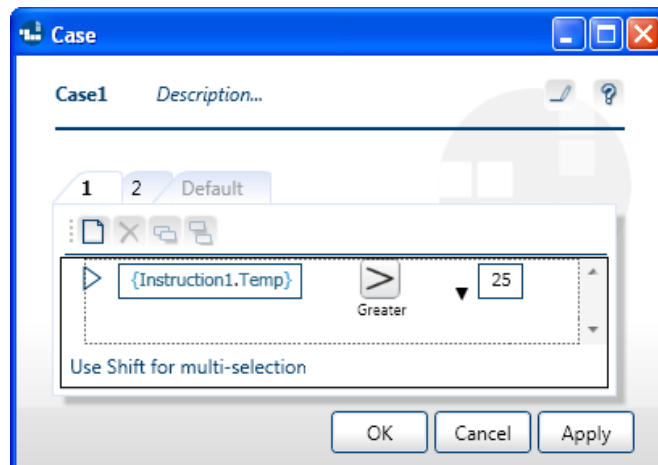
Figure 8 - Set the "Too Hot" condition

m) Select the second conditions (tab 2). Use the same variable (either type or insert), change the condition sign to smaller and the value to 20

n) From the utilities stencil, add a *MESSAGE* box to each condition flow (see Figure 9 - adding a message to the "TOO HOT" flow)

o) Edit each message with the relevant text – for the too hot condition, add the text "PLEASE ACTIVATE COOLER", the too cold "PLEASE ACTIVATE HEATER" and "TEMPERATURE OK" for the default case.
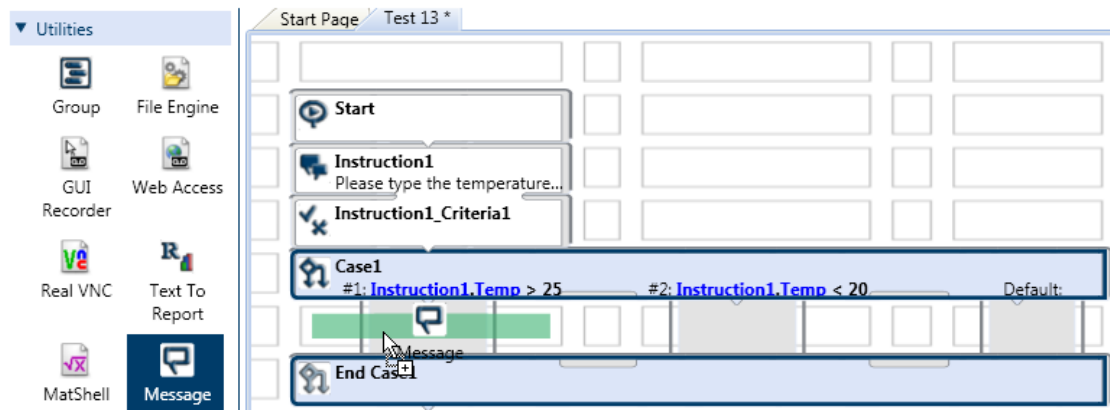


Figure 9 - adding a message to the "TOO HOT" flow

p) The final test would look like Figure 10 - The finished test
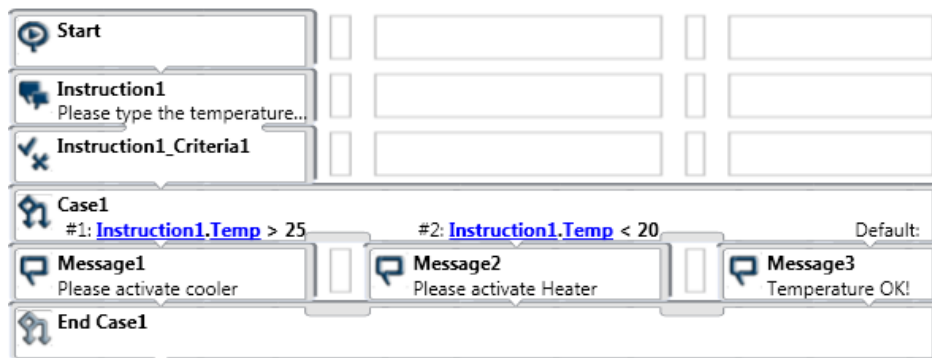
Figure 10 - The finished test



Figure 11 - Edit a message

q)  Run the test by clicking on ▷ Run ,on the ribbon

# The Work Environment
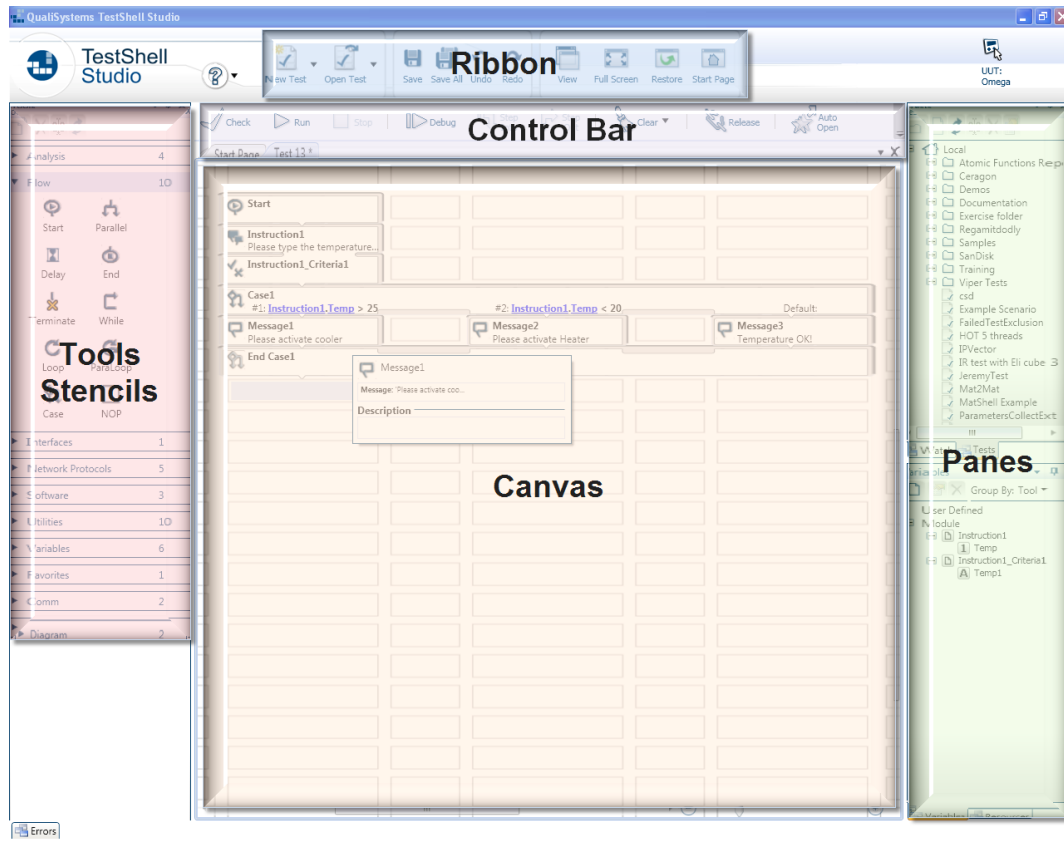


## 1.1     **Ribbon**

General actions in documents: Opening, saving, clearing and window options

## 1.2     **Control Bar**

Used to 'Dry Run' a test in Studio, before launching on a runner.

## 1.3     **Tools Stencils**

The repository of modules. From here, one drag a step onto the canvas

## 1.4     **Panes**

The different explorers (Tests, UUT, Regression, watchers, etc.)

## 1.5     **Canavs**

The document authoring area, place steps and constructs to create flows.

# Glossary

## 1.1  Assets

A way to handle testing resources like scripts, DLL's executables and web services. Shared assets are saved on the Database and are deployed on each runner temporarily.

## 1.2  Attributes

Any measurement, and many modules, can be tagged with attributes. Attributes are tags of metadata that will allow the aggregation, filtering and grouping of data in the database. Examples of attributes could be: Versions (HW/SW/FW), Test conditions, Environmental data, report sections indication, etc.

## 1.3  Canvas

The work area of a document. Canvases are made from blocks and separators. Place steps and constructs on a canvas to create a flow.

## 1.4  Construct

Some elements (Loops, Cases, parallel, lock) made from two steps – a beginning and a rendezvous step. One needs to place other steps inside the construct, as needed.

## 1.5  Criteria

A decision step. Criteria steps decide if the result passes or fails according to a set of conditions. It takes just one failed criteria (or a *FAIL* step) to change a report status to *FAIL*. Criteria can be added to any measurement module, or to be placed autonomously on the canvas, as a step.

## 1.6  Document

A collective name to everything that has a canvas (Test, Regression, Group, mini-canvass, function, etc.)

## 1.7  Event

A special kind of variable, allows one flow to time the activation of a parallel flow.

## 1.8    **Flow**

All actions in a document, start with a *START* step and flow down (and right, if needed) until no more steps are available, or reaching an *END / TERMINATE* steps. The single cascade of steps is called a flow. A document can have several flows 9either have multiple *START* steps, or branch with *PARALLEL*)

## 1.9    **InSight®**

The Business Intelligence tool of the TestShell suite. Used to query on results from Studio® Runners to create reports and web dashboards. Attributes can be used by InSight to aggregate, filter and group results in graphs, charts, OLAP cubes and web parts.

## 1.10    **Mini-Canvas**

Some modules (e.g. communications) are opened on a separate document, to create a session. This is needed as these session include multiple steps (Setting user name and passwords, selecting a long list of options, etc)

## 1.11    **Module**

Any element that can be placed on a canvas to become a step or construct. includes tools, assets, variables and tests.

## 1.12    **Pane**

A window element that appears on the screen, that can be made into a floating or docked element (Watcher, Tests, Regressions, Variables, Assets, Tools, Errors, Sessions, etc.)

## 1.13    **Publishing Variables**

Variables in a document can be published as inputs and or outputs. Once published, a variable is accessible from outside a document. One can set input variables and read output variables, when a test is used by another test as a function

## 1.14    **Regression**

A document that combines several tests to fully examine a device, service or a product.

## 1.15    **Ribbon**

A set of buttons that appear on the top of the work area and are always visible

## 1.16 **Step**

Any assigned block on the canvas. Steps can be modules or part of a construct.

## 1.17 **Test**

A document which does a specific thing. Tests consist of at least one flow, and can publish variables as inputs and or outputs, which allow them to be functions in other documents.

## 1.18 **Transformation**

An analysis step. Transformation allows parsing, calculations, conversions and more. Transformation can be added to any measurement module, or to be placed autonomously on the canvas, as a step.

## 1.19 **UUT**

Unit Under Test (Or DUT – for Device Under Test). A grouping element to tag tests and regressions execution. Each UUT can define parameters that all documents can use as global variables.

## 1.20 **Variable**

Studio®'s most basic infrastructure. A variable is a cell (or a group of cells) in memory, which can store and retrieve information. Variables can be string, numeric, event, lock or session and can be scalar, vector or a matrix of values. Variable are used throughout Studio®.

## 1.21 **Watchers**

Watchers appear in the watcher pane, they display in real time the values of variables (or steps) beside the canvas. Used mainly for debugging or monitoring.

## Modules

| Icon | Description | Tool Group |
|------|-------------|------------|
| **Pass** | **Adds a "Pass" notification to a test report.** | **Analysis** |
| **Fail** | **Adds a "Fail" notification to a test report. Will cause test status to turn "Fail"** | |
| **Criteria** | **Determines a "Pass" or "Fail" according to a set of conditions. This is a manifestation of the DECISION step in testing theory** | |
| $T_{(i)}$ **Transform** | **Converts, parses, transforms and manipulates numeric or textual information. This is a manifestation of the ANALYSIS step in testing theory** | |
| **Attributes** | **Adds metadata (or tags) to measurements or entire tests. Attributes are very useful to aggregate, group and filter data in BI applications (like QualiSystems InSight©)** | |
| **R** **Text To Report** | **Adding text, variables, expressions, files, links and images to the report** | |
| **Start** | **An obligatory step in every document. Each 'Start' step will start an independent flow on the document (test, group or regression)** | **Flow** |
| **End** | **Three different verities:** <br> **Stop branch – will stop the flow, but not any parallel flow** <br> **Stop Test – will stop all flow in current document, but not any parallel tests** <br> **Stop Execution – will stop all running documents.** | |

| | | |
|---|---|---|
| **C** Loop | **Repeating a flow of steps for a set number of times.** | **Flow** |
| **Terminate** | **Will stop all execution, and will change report status to "Terminated". Use only for emergency stops (not regular test condition).** | |
| **Parallel** | **Branch the flow to two (or more) parallel flows** | |
| **ParaLoop** | **run a segment of the flow several instances concurrently** | |
| **Delay** | **Pause a flow to a specific set of time** | |
| **While** | **Similar to loop. This construct repeats a segment of a flow as long as a condition is met** | |
| **Case** | **Selecting a flow according to a set of conditions.** | |
| **NOP** | **No Operation** | |
| **GUI Automator** | **Recording and playing windows and web interaction. Use with a Ranorex Installation (Independent License)** | **GUI Automation** |
| **Real VNC** | **Using a Real VNC remote client/server to perform actions on the remote machine or phone. Use with a Real VNC agent (Independent License)** | |
| **VISA** | **Communications with a VISA mapped device (Serial, GPIB, LXI, VXI, etc.)** | **Interfaces** |
| **SNMP** | **Simple Network Management Protocol** | |

| | | |
|---|---|---|
| **SNMP Trap** | **Sets, reads and closes SNMP traps (with filters)** | **Network protocols** |
| **FTP** | **An Files Transfer Protocol client** | |
| **Terminal** | **A 'mini-canvas' module for a session of Serial, Telnet or SSH communications** | |
| **Client** | **A 'mini-canvass' module to set and monitor the communication on Serial, TCP and UDP protocols** | |
| **LabVIEW VI** | **Integrating National-Instruments© LabView® Virtual Instruments** | **Software** |
| **Win32 DLL** | **Integrating a win32 DLL (C or Winapi)** | |
| **Command Shell** | **Using the Microsoft© PowerShell for DOS or simple scripting** | |
| **Group** | **Create an empty group (a sub test) within a test** | **Utilities** |
| **File Engine** | **File actions (creating, loading, copying, renaming, searching, etc.)** | |
| **MatShell** | **A mathematical scripting tool** | |
| **Message** | **A pop up message – for warning, instructing, guiding and alerting a user** | |
| **Instruction** | **Getting information from a user** | |

| | | |
|---|---|---|
| **Test** | **using another test as a function within the current document** | **Utilities** |
| **X=** Assign Variable | **Set a variable to a fixed or calculated value(s)** | **Variables** |
| **Set Session** | **Setting a session variable to activate a session** | |
| **End Session** | **Setting a session variable to stop (kill) a session** | |
| **Lock** | **A mutual exclusive construct. This will make a segment of flow to be exclusive from parallel *LOCK* segments** | |
| **Set Event** | **Setting an Event type variable to either *ON* (default) or *OFF*. This will trigger the corresponding *WAIT FOR EVENT* modules** | |
| **Wait For Event** | **Pausing a specific flow until a *SET EVENT* (above) is triggered in a parallel flow** | |

| Icon | Description | Tool Group |
|------|-------------|------------|
| **Write** | **Send information using the selected protocol** | **Network Tools (Mini Canvas)** |
| **Read** | **Collects the receive buffer of the selected protocol** | |
| **Terminal Command** | **Combination of Write and Read. The module will send information and collect the response** | |

| Icon | Description | Tool Group |
|------|-------------|------------|
| **Inspect** | **Measuring system elements and allow Javascript execution** | **GUI Tools (Mini Canvas)** |
| **Keyboard** | **create a key press sequence** | |
| **Mouse** | **Create mouse operations on selected elements** | |
| **Validate** | **Validating element property values** | |

## Running Tests and Regressions

To run a test or regression in the RUNNER environment, please follow the following steps:

1.         Click on the Runner Icon

        (Alternative: Start Programs->QualiSystems TestShell->TestShell Runner)

2.         After the splash screen finish load, type your credentials

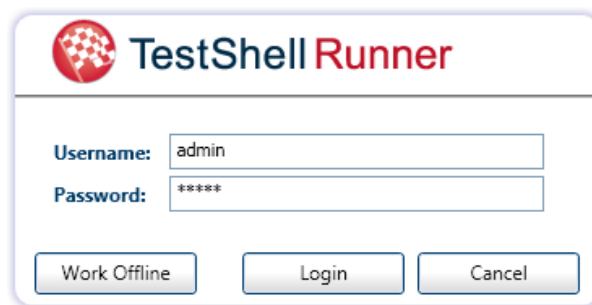        (by default - User: admin Password: admin)

Figure 12 - The Runner Login

3.         Select the test (or regression) to run by clicking on the *OPEN TEST* or *OPEN REGRESSION* buttons on the ribbon (To switch, click in the black triangle near the button )

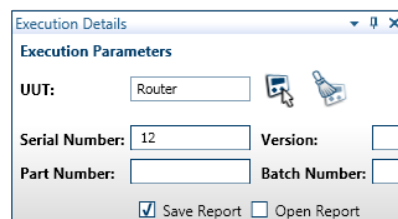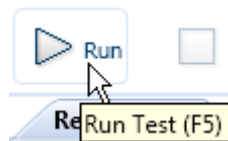4.         Once the document loads, fill the execution details:

Figure 13 - Execution details

To run the document, click on the RUN button in the control bar (or click F5)