# StepWise: Concept and Features

*StepWise* is a Windows based test automation tool. What makes it different from other standard tools (including *QTP*) is that it helps in partially automating test scenarios. Some examples for scenarios that need partial automation are synchronizing the test execution, verification procedures that often need expensive investment to create a fool-proof automation around it, handling of exceptions and so on. In all the above mentioned cases, manual intervention is desirable. In the proposed scheme of partial automation, there should be a mechanism that seamlessly switches between the manual and automated execution.

To start with, the tool helps you at creating the test scripts, step by step with the help of a wizard. Content provided by the wizard is easily manageable.

Apart from helping with partial automation of test scenarios, *StepWise* helps you at debugging your test scripts. Thus, you can step into each line in your script, treading carefully rather than executing it in normal mode, wherein the script is executed in one-go. In the normal mode, execution always occurs sequentially, beginning with the first line to the last. The tool will also allow you to skip (using breakpoints) certain portions of the test and also execute selected parts of the script. If you notice that the execution is going awry due to specific environmental situations, you could quit the test. Similarly, at any point in time during execution, you can interrupt and key in your observations. All such comments will be time stamped and logged, that can be used for analysis later on.

The tool logs everything that occurs during test execution in *debuglog.txt*. *debuglog.txt* is a running log, and works across scripts and different runs. In addition to it, the tool also maintains a *per-script run* log that logs only one line per step. Upon encountering an error - either due to script failure, or an environment issue (including the product under test), the tool will alert you with the help of balloons popping up in the system tray. This is in addition to the logs.

Test scripts can be executed using either the *StepWise* tool interface, or from a command-line interface. It might be simpler to execute from a command interface, because in this case you wouldn't need to install the tool (or the .NET framework), but the execution engine (*automate.exe*).

Finally, the tool will help you at creating a very re-usable, reliable and extensible test framework, through a very intuitive interface. With this process, you'll essentially cut down more than 50% overhead on framework creation.

## FAQ

### 1. What are the pre-requisites to run the tool?

.NET framework 2.0 is required to run the *StepWise* tool. To execute, you only need to have copied 'Engine/automate.exe' to your computer.

In case the tool doesn't run even with .NET present, follow these steps.

a. Create a file in the same folder as the tool. Name it *StepWise5.8.exe.config*.

b. Type the following text into the file and save it.

<configuration>

<startup>

<supportedRuntime version="v4.0" />

</startup>

</configuration>

c. Before you can rerun the tool, create one more file called *Record.exe.config*, type in the same content as above and save it.

d. Re-run the tool.

### 2. How do execute a test from command-line?

*automate.exe ..\scripts\<script-name>*

NOTE: You can optionally use the following switches during CLI execution : -debug (start execution in debug mode), -nobreaks (ignore breaks during executions), -batch (batch mode)

Use with –help switch if you want the help.

### 3. How do I create a test script?

a. The first step is to understand the test scenario completely. Once you've done that, identify the methods you'll use. Most of the GUI-driven testing can be done with the help of a handful of methods provided by the tool - *Run*, *WinWaitActive*, *WinActivate*, *WinClose*, and *Record*. *Record* will let you record any actions you'll perform using a keyboard and a mouse, and replay. Thus, in the case of GUI-driven testing, the recommended best best course to take is to use Run to start the application, and record the rest. In the case of a non-GUI based testing, make use of the other methods supplied with the tool. In case these methods do not serve your purpose entirely, create your own methods and add to the list of available methods (explained in another section).

b. Once you've finished recording, revisit the recorded script and tweak/remove it as per your requirements. For example, since you'll typically not want to rely on mouse movements due to its inherent unreliable nature, remove all lines that indicate mouse-clicks and drags.

c. Do not finalize your script until you've executed it at least once. Ideally, you should try it a different machine (than the one it's recorded in) before finalizing.

### 4. How do I add a method into a test script?

a. Select the category. Categories can be sub-divided, so search for the appropriate one.

b. Select the method.

c. Provide necessary parameters and hit 'Add Step'.

### 5. Can I create short-cuts to my scripts?

Yes. Here’s how.

a. Once you create a test script, save it to the default location.

b. Once it's saved, hit 'Refresh' on the side of the 'Script' combo-box. The script should now be listed in the combo.

### 6. Results file do not get created. Why?

a. Result (.PASS/.FAIL) gets created only in the case of a saved test script. So, save your script and then execute. Again, hit 'Refresh' on the side of the combo.

### 5. My test script is really large. How can I manage it better?

Create a UDF (user defined function) out of your script.

a. Hit 'Save As UDF' menu item.

b. Provide the method name you want to save the script as. Do not use spaces or special characters here.

c. Follow the instructions in the last dialog.

### 6. Keyboard/mouse seems to be unreliable in my scenario. Is there an alternative?

Yes, if you're working with standard windows controls. Try the methods available under 'Common-Components-Basic' category.

When it asks for 'Control name', use the *Window info* tool available under *Tools* menu and look for *ClassnameNN*. Double-click, copy the value and paste it in the appropriate edit-box.

### 7. My script is very lengthy. Can I execute part of it?

Yes. Here’s how.

a. Select the part you want to execute, right-click and select 'Run this block'. Alternatively, use 'Run from here' or 'Run to here' options.

### 8. I finalized my script, but I discovered later that a particular part of the script does not work. What can I do?

Remove the part that's not working. Place your cursor at the appropriate portion in the test script. Re-record or select another method and hit 'Add step'.