HILEVEL Technology, Inc.

Q'n Apps #S56

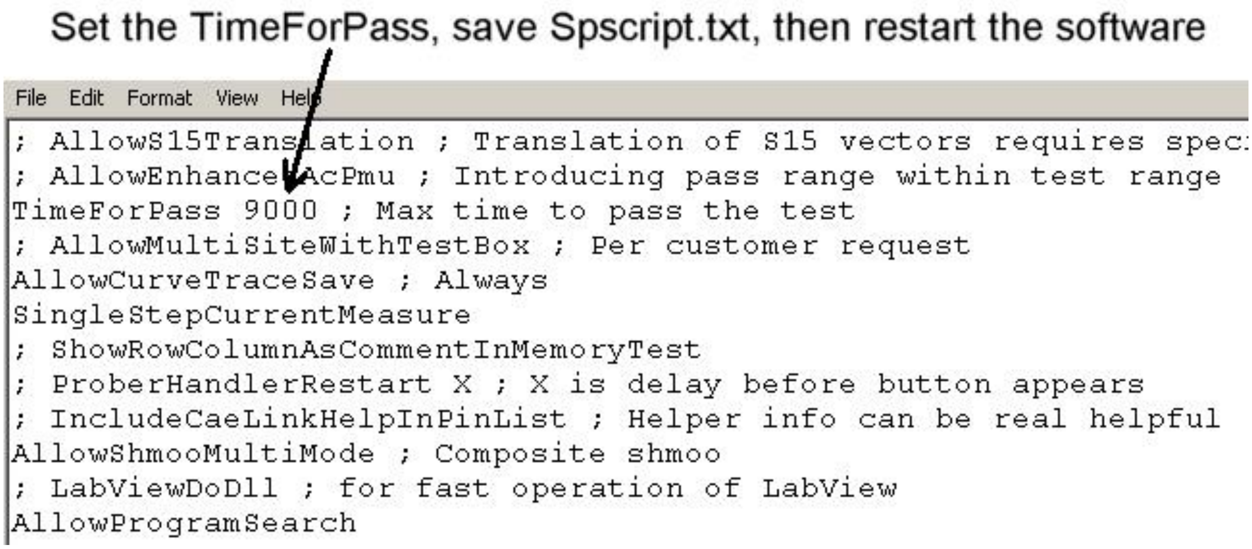
Q: Why do I get Timeout errors in AutoTest?

Setting the Time For Pass

Some devices, like EPROMs or FLASH memories, require a long time for “burning” the program data into the device. This is usually accomplished with loops in the programmable pattern generator, achieving the necessary milliseconds of program time. If this needed time is quite long, the user may get a Timeout error when running the test from AutoTest. All you need to do is set a longer Timeout period, and HiLevel has two methods for doing this.

Where is the Timeout value stored?

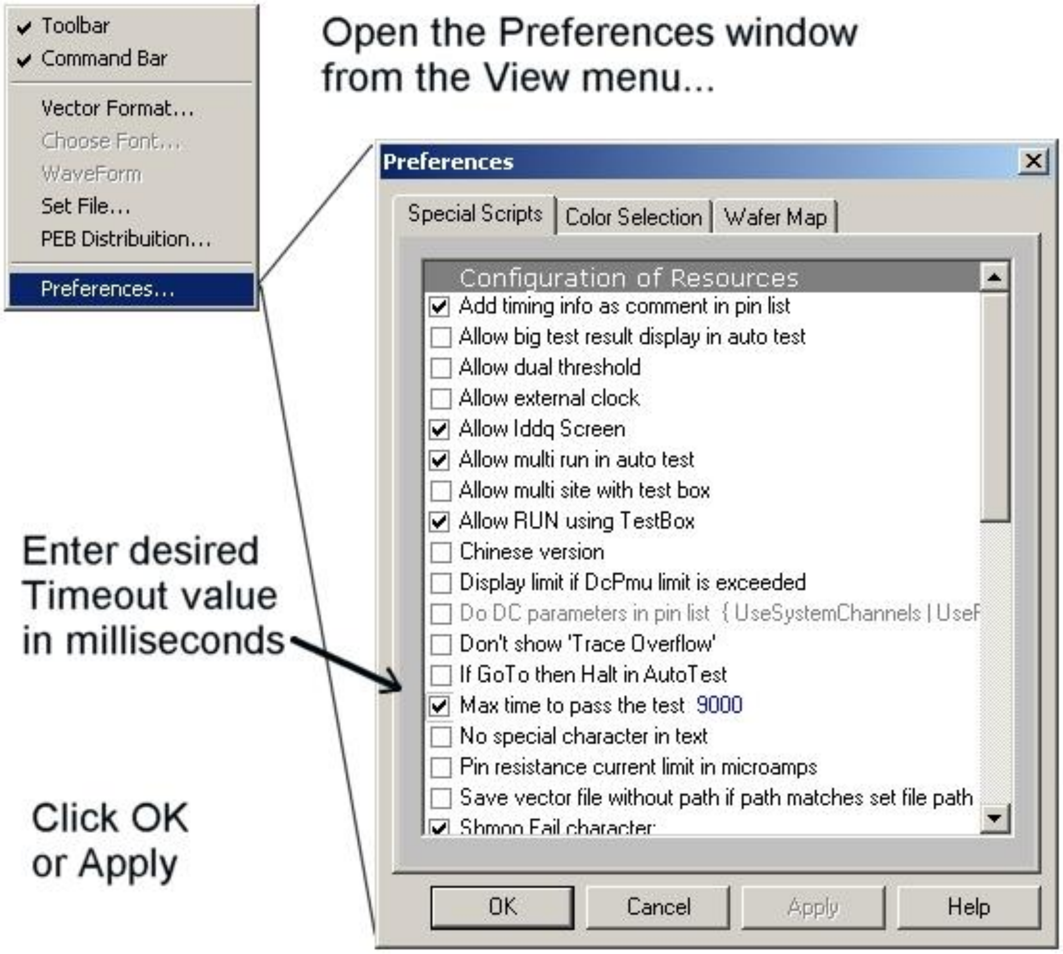
The Timeout value (“Time For Pass”) can actually be stored in three places. The primary location for this is in the Special Scripts file, Spscript.txt, located in your Symphony folder. Whenever you start Symphony, the keywords in Spscript are loaded into the software to enable certain features. You can set the value in this file, and whenever you start to build a new setfile this will be the default Timeout value.


Set the TimeForPass, save Spscript.txt, then restart the software

```
; AllowS15Translation ; Translation of S15 vectors requires spec
; AllowEnhancedAcPmu ; Introducing pass range within test range
TimeForPass 9000 ; Max time to pass the test
; AllowMultiSiteWithTestBox ; Per customer request
AllowCurveTraceSave ; Always
SingleStepCurrentMeasure
; ShowRowColumnAsCommentInMemoryTest
; ProberHandlerRestart X ; X is delay before button appears
; IncludeCaeLinkHelpInPinList ; Helper info can be real helpful
AllowShmooMultiMode ; Composite shmoo
; LabViewDoDll ; for fast operation of LabView
AllowProgramSearch
```

Example of Spscript.txt file

Another place where the Timeout is stored is in your setfile. This means that every setfile can have a different timeout value. So when you load a setfile, the Timeout is changed in the system to the value in the setfile. If you have already created a setfile and the Timeout is too small, you simply change the Timeout within the software and then re-save the setfile. This must be done to use the Timeout in AutoTest.



Open the Preferences window from the View menu...

Configuration of Resources

☑ Add timing info as comment in pin list
☐ Allow big test result display in auto test
☐ Allow dual threshold
☐ Allow external clock
☑ Allow Iddq Screen
☑ Allow multi run in auto test
☐ Allow multi site with test box
☑ Allow RUN using TestBox
☐ Chinese version
☐ Display limit if DcPmu limit is exceeded
☐ Do DC parameters in pin list { UseSystemChannels | UseF
☐ Don't show 'Trace Overflow'
☐ If GoTo then Halt in AutoTest
☑ Max time to pass the test 9000
☐ No special character in text
☐ Pin resistance current limit in microamps
☐ Save vector file without path if path matches set file path
☑ Shmoo Fail character

Enter desired Timeout value in milliseconds

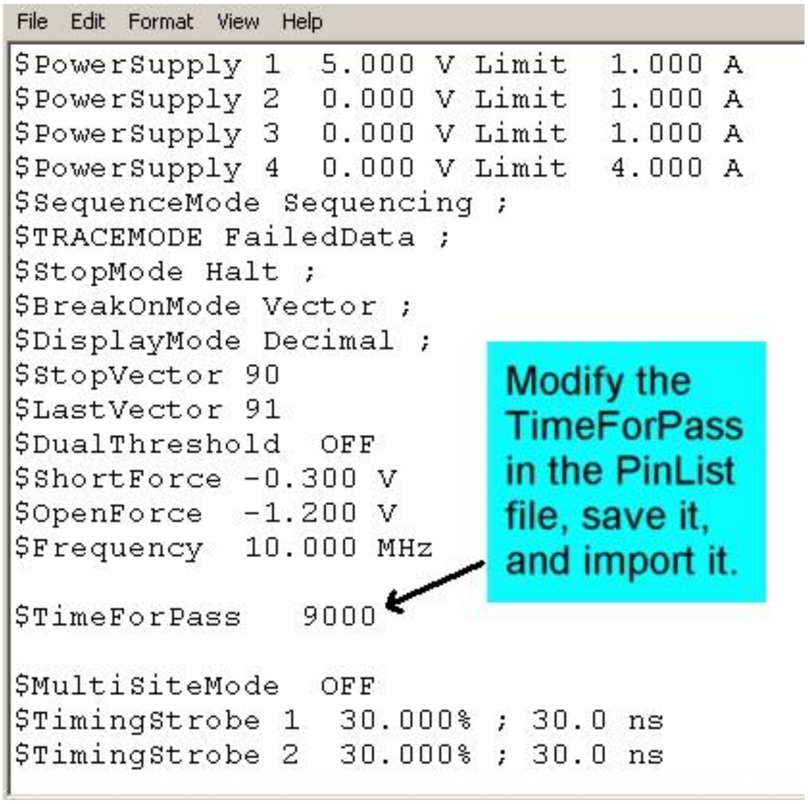
Click OK or Apply

Setting the Timeout from the Preferences window

And finally, you can also change Timeout for a specific test setup by modifying your PinList file. Simple create a PinList file, open it, and change the value. But remember, you must save your setfile after importing the PinList to use the new Timeout in AutoTest. See an example of the PinList method on the next page.

Why Have a Timeout at All?

Or why not just set it to some huge value? In Production, the goal is to test as many parts per minute as possible. Some tests may contain such things as "pattern matching", where the test stays in a loop until a particular pin changes state. If the device is bad, it may never get the pattern match and the test could hang because it is waiting for the match. Without a Timeout, users would need to monitor the tests all the time, and that is not what we want on an automated test floor. We want to set the Timeout to a reasonable value that makes sense for each test of each device in order to optimize throughput.



```
$PowerSupply 1  5.000 V Limit  1.000 A
$PowerSupply 2  0.000 V Limit  1.000 A
$PowerSupply 3  0.000 V Limit  1.000 A
$PowerSupply 4  0.000 V Limit  4.000 A
$SequenceMode Sequencing ;
$TRACEMODE FailedData ;
$StopMode Halt ;
$BreakOnMode Vector ;
$DisplayMode Decimal ;
$StopVector 90
$LastVector 91
$DualThreshold  OFF
$ShortForce -0.300 V
$OpenForce  -1.200 V
$Frequency  10.000 MHz

$TimeForPass   9000

$MultiSiteMode  OFF
$TimingStrobe 1  30.000% ; 30.0 ns
$TimingStrobe 2  30.000% ; 30.0 ns
```

Modify the TimeForPass in the PinList file, save it, and import it.

Setting the Timeout in a PinList file

Also See:
Q'nApp #S1: PinList
Q'nApp #S3: AutoTest
Q'nApp #S12: AutoTest
Q'nApp #S15: Special Script
Q'nApp #S16: Symphony Files