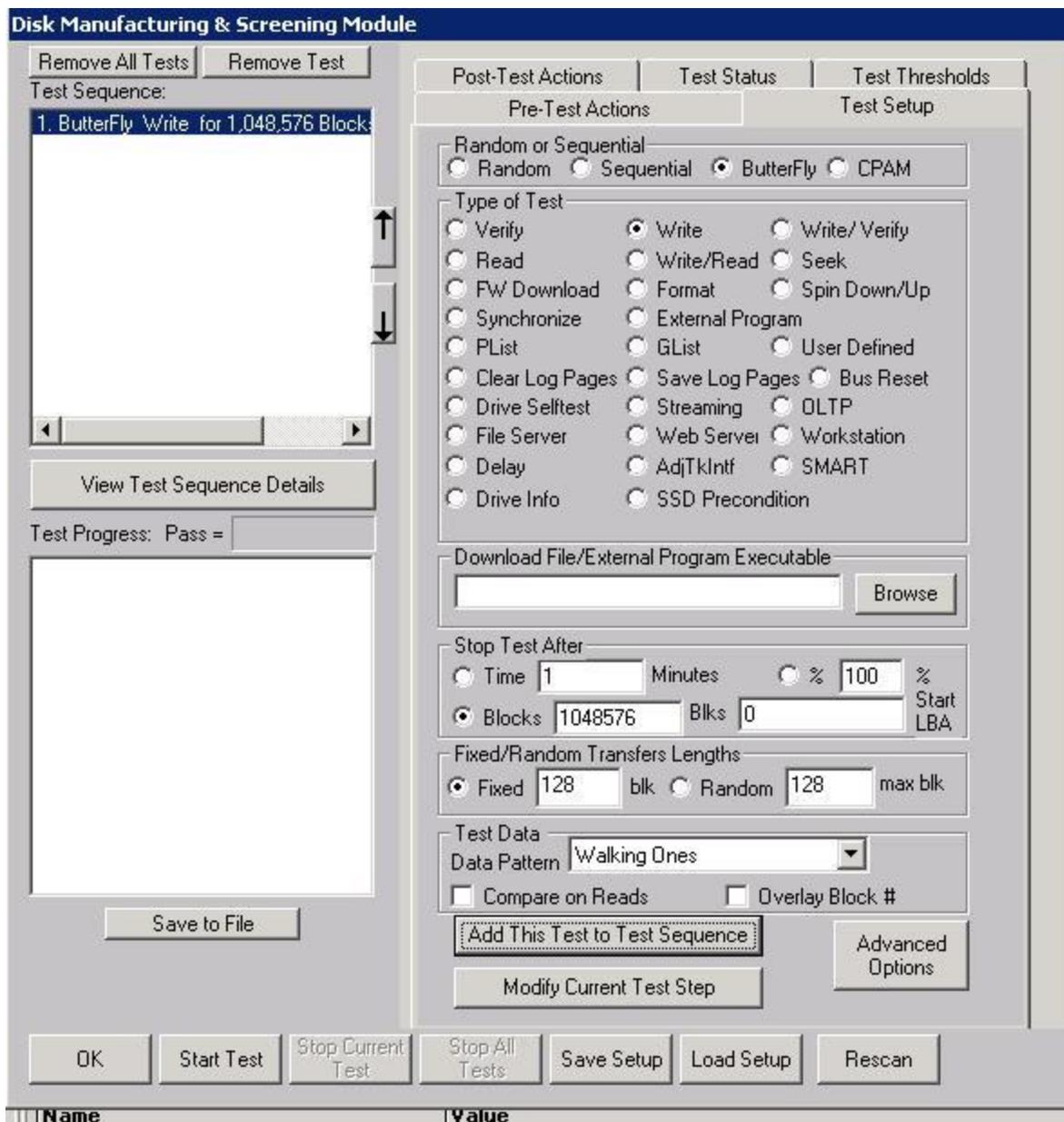# Automated Manufacturing Engine

The Automated Manufacturing Engine (AME) is STB's new automation tool that allows you to run scripts created in our Disk Manufacturing Module (DMM). AME can be run from a Command Prompt Window or even a batch file. Like other automation tools, once run it requires no user-intervention or "baby sitting".   After AME completes, the user can then analyze the extensive logfiles for success or failures.

In order to run AME, there are four steps that must be done. These steps are

1. Create, within DMM, one or more testing scripts
2. Decide which devices on your system will be tested
3. Create a configuration file that will test the devices in step 2
4. Launch AME

STEP 1: Create, within DMM, one or more testing scripts

Our first script that we will create in DMM will consists of 4 tests. The first test is a "Write Test, Butterfly Access, for 1048576 blocks". See the pic below:

Disk Manufacturing & Screening Module

Remove All Tests | Remove Test

Test Sequence:

1. ButterFly Write for 1,048,576 Blocks

View Test Sequence Details

Test Progress: Pass =

Save to File

| Post-Test Actions | Test Status | Test Thresholds |
| Pre-Test Actions | | Test Setup |

Random or Sequential
○ Random ○ Sequential ● ButterFly ○ CPAM

Type of Test
○ Verify   ● Write   ○ Write/ Verify
○ Read   ○ Write/Read ○ Seek
○ FW Download ○ Format ○ Spin Down/Up
○ Synchronize ○ External Program
○ PList ○ GList ○ User Defined
○ Clear Log Pages ○ Save Log Pages ○ Bus Reset
○ Drive Selftest ○ Streaming ○ OLTP
○ File Server ○ Web Server ○ Workstation
○ Delay ○ AdjTkIntf ○ SMART
○ Drive Info ○ SSD Precondition

Download File/External Program Executable
[   ] Browse

Stop Test After
○ Time [1] Minutes   ○ % [100] %
   Start
● Blocks [1048576] Blks [0]   LBA

Fixed/Random Transfers Lengths
● Fixed [128] blk ○ Random [128] max blk

Test Data
Data Pattern [Walking Ones ▼]
☐ Compare on Reads   ☐ Overlay Block #

Add This Test to Test Sequence   Advanced Options

Modify Current Test Step

OK | Start Test | Stop Current Test | Stop All Tests | Save Setup | Load Setup | Rescan

Name | **Value**

Our second test will be a "Read Test with Data Compare, Butterfly Access, for 1048576 blocks" – this step is to verify that the pattern "Walking Ones" was indeed written to the start and end of the drive. Our third step is an "External Program Test". The "External Program" name is the DMM's "WriteSameScsi" Test (you do not need, in this case, to create this executable, DMM already "knows" about it). See the pic below:
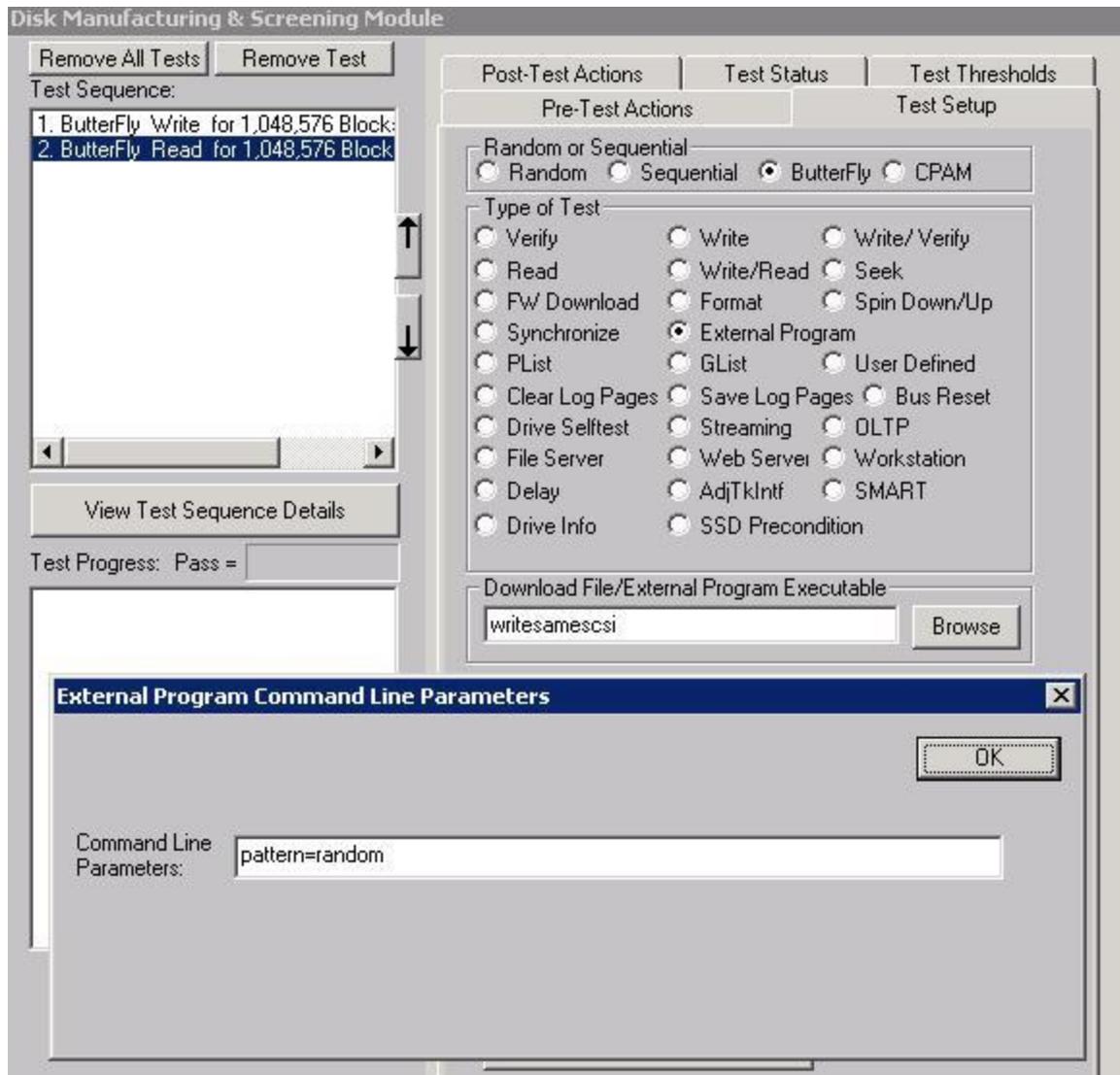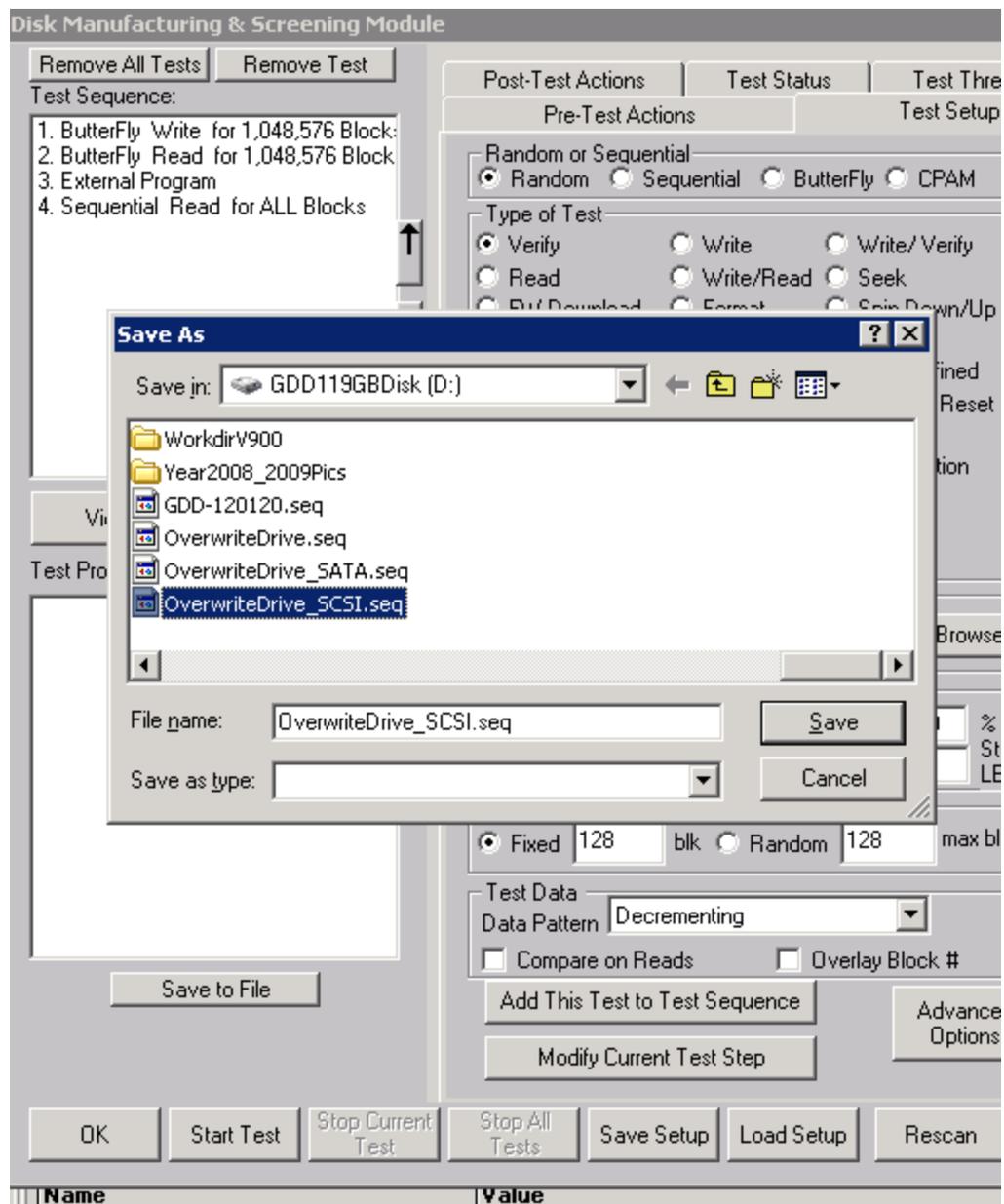
Notice in the pic above, as soon as we clicked the radio button "External Program" DMM brought up a dialog box asking for the command line parameters. In the above pic, we have entered "pattern=random" (this tells the WriteSameScsi application to write a random pattern to the drive).

Our final test is a "Read Test, Sequential Access, for the Entire Drive" – this step is to verify the entire drive has in fact been written with a random pattern.
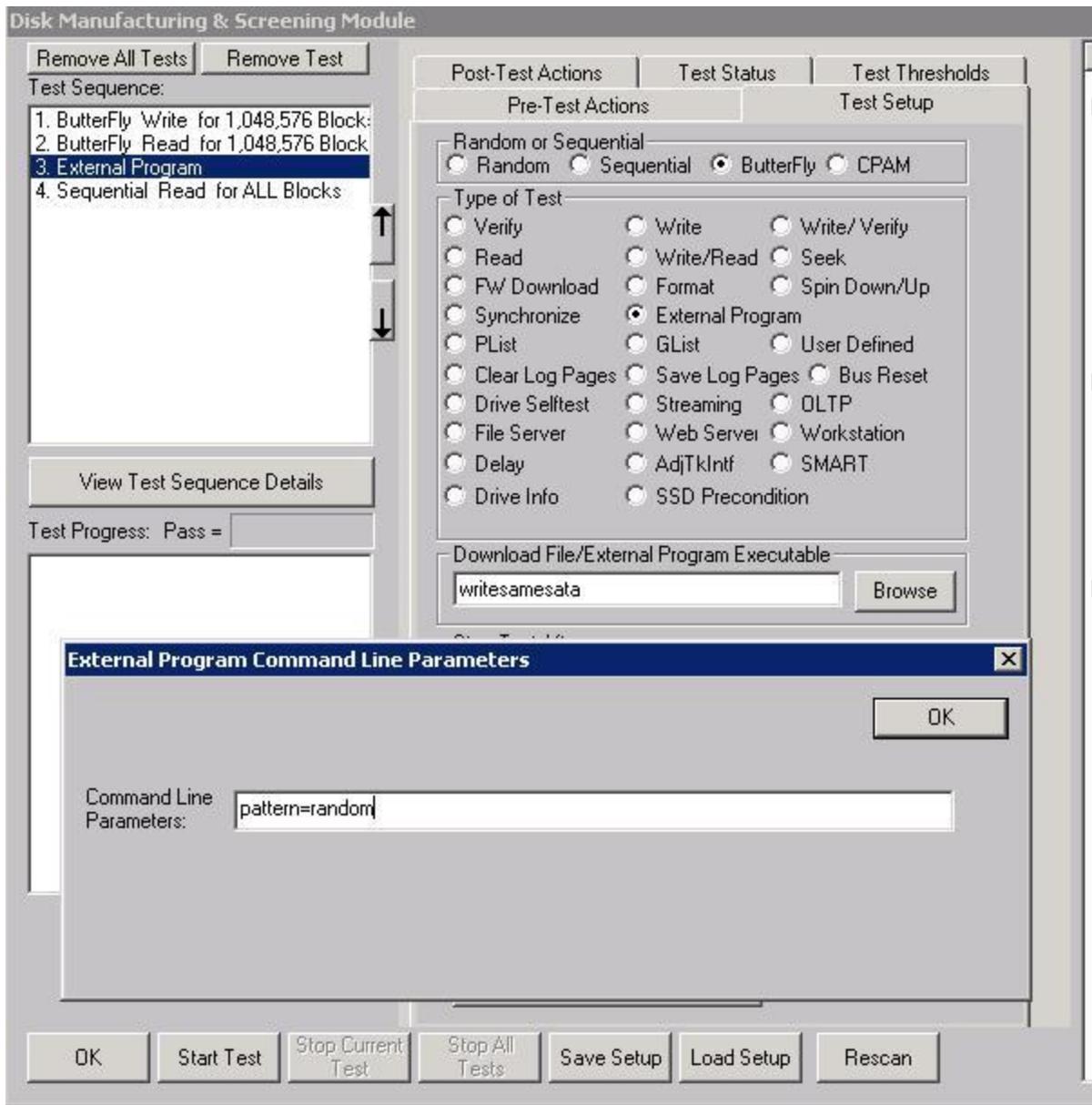
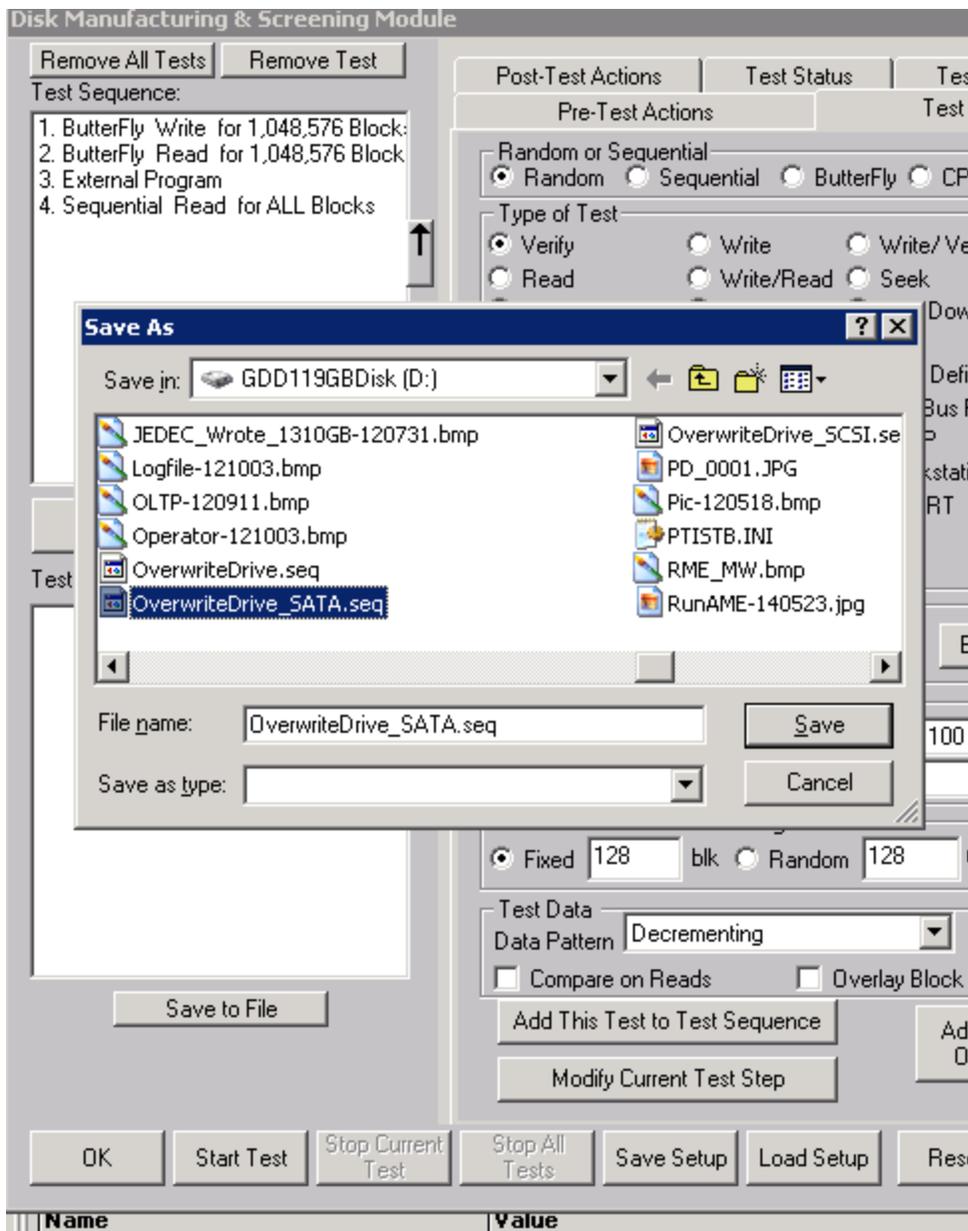In DMM, the 4 tests in our script looks like what is shown in the below pic:
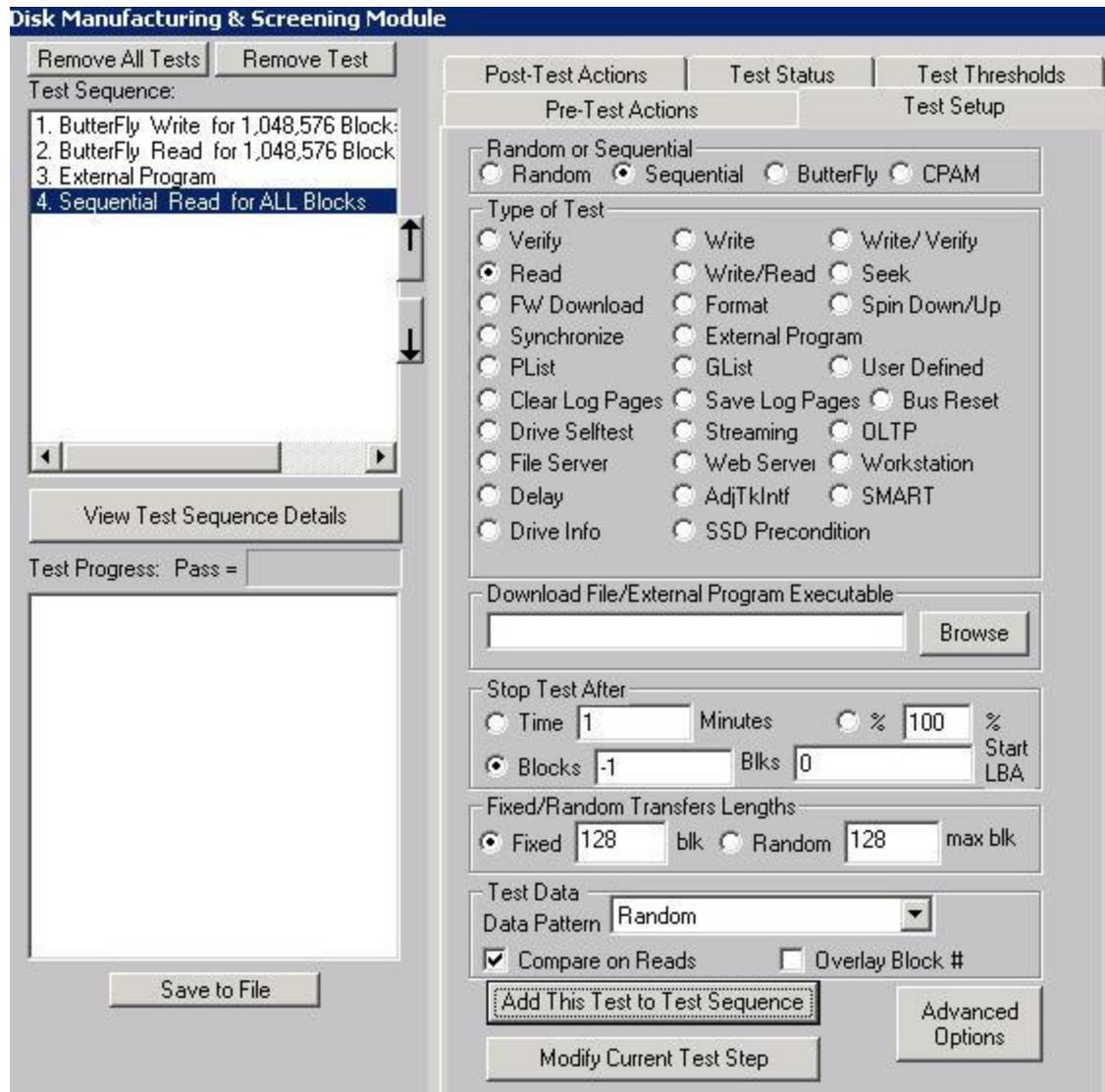
Now let's save off this script – this script will be "fed into" our AME product. Go ahead and click the "Save to File" button, and save it off to the file "OverwriteDrive_SCSI.seq" (see the pic below):

Disk Manufacturing & Screening Module

Remove All Tests | Remove Test

Test Sequence:
1. ButterFly Write for 1,048,576 Block:
2. ButterFly Read for 1,048,576 Block
3. External Program
4. Sequential Read for ALL Blocks

Post-Test Actions | Test Status | Test Thre

Pre-Test Actions | Test Setup

Random or Sequential
⦿ Random ○ Sequential ○ ButterFly ○ CPAM

Type of Test
⦿ Verify ○ Write ○ Write/ Verify
○ Read ○ Write/Read ○ Seek
○ DL/ Download ○ Format ○ Spin Down/Up

Save As

Save in: 📀 GDD119GBDisk (D:) ▾ ← 🔼 📂 ▦▾

📁 WorkdirV900
📁 Year2008_2009Pics
📄 GDD-120120.seq
📄 OverwriteDrive.seq
📄 OverwriteDrive_SATA.seq
📄 **OverwriteDrive_SCSI.seq**

File name: OverwriteDrive_SCSI.seq | **Save**

Save as type: ▾ | Cancel

⦿ Fixed 128 blk ○ Random 128 max bl

Test Data
Data Pattern Decrementing ▾
☐ Compare on Reads ☐ Overlay Block #

Save to File

Add This Test to Test Sequence | Advance Options

Modify Current Test Step

OK | Start Test | Stop Current Test | Stop All Tests | Save Setup | Load Setup | Rescan

Name | **Value**

Now let's create a second script – this script is identical to the above script except that it is designed for SATA drives. Test steps 1,2, and 4 are identical, but step 3, the "External Program Test" is different in that the program name now is "WriteSameSATA" (see the pic below):

Disk Manufacturing & Screening Module

Remove All Tests | Remove Test

Test Sequence:
1. ButterFly Write for 1,048,576 Block:
2. ButterFly Read for 1,048,576 Block
3. External Program
4. Sequential Read for ALL Blocks

View Test Sequence Details

Test Progress: Pass =

Post-Test Actions | Test Status | Test Thresholds
Pre-Test Actions | Test Setup

Random or Sequential
○ Random ○ Sequential ● ButterFly ○ CPAM

Type of Test
○ Verify   ○ Write   ○ Write/Verify
○ Read   ○ Write/Read   ○ Seek
○ FW Download   ○ Format   ○ Spin Down/Up
○ Synchronize   ● External Program
○ PList   ○ GList   ○ User Defined
○ Clear Log Pages ○ Save Log Pages ○ Bus Reset
○ Drive Selftest   ○ Streaming   ○ OLTP
○ File Server   ○ Web Server   ○ Workstation
○ Delay   ○ AdjTkIntf   ○ SMART
○ Drive Info   ○ SSD Precondition

Download File/External Program Executable
writesamesata | Browse

External Program Command Line Parameters   ✕

OK

Command Line
Parameters: pattern=random

OK | Start Test | Stop Current Test | Stop All Tests | Save Setup | Load Setup | Rescan

Go ahead and save this script off to file "OverWriteDrive_SATA.seq"

STEP 2: Decide which devices on your system will be tested

AME has two ways to tell it which devices you wish to test. These are the

"device="

"hba="

The two parameters that you will pass to AME tell it which devices to test (more on this in STEP 4). But here are a few examples of using these two parameters:

Example 1: AME.exe AMEConfig.txt device=3-5-0,6-0-0

In this example, AME will test devices 3-5-0 (HBA=3, Target=5, Lun=0), and device 6-0-0.

You can have as many devices as you want listed. Notice they must be separated by a comma, and that there are no spaces in between.

Example 2: AME.exe AMEConfig.txt hba=3,6

In this example, AME will test all devices attached to HBA #3 and HBA #6.

You can have as many HBAs listed. Notice the list of HBA numbers are separated by a comma, and that there are no spaces in between

Example 3: AME.exe AMEConfig.txt hba=3,6,device=4-1-0

In this example, AME will test all devices on HBA #3 and #6, and also device 4-1-0

STEP 3: Create a configuration file that will test the devices in step 2

AME requires a configuration file to be passed as the first parameter (see the above examples). This configuration file has two very important functions:

1. Informs AME where the logfiles are to be stored
2. Informs AME which scripts are to be run against which devices

To inform AME where the logfiles are to be stored, you use the keyword "PATHTOLOGFILES". Here's an example:

PATHTOLOGFILE=d:\workdirv900\amelogfiles

IMPORTANT: We recommend that there be NO spaces in the path to the logfiles!

To inform AME which scripts are to be run against which devices, we use the keyword "DEVSEQ". Here's an example:

DEVSEQ:VENDOR=Seagate,SEQ=OverWriteDrive_SCSI.seq

Notice in the above example, the format is basically

DEVSEQ:VENDOR=abc,SEQ=xyz.seq

What this DEVSEQ line in your configuration file tells AME is that out of all the devices you list in STEP 2, any one of them which is made by Seagate, AME will run test sequence file "OverWriteDrive_SCSI.seq" against these drives.

Here's another example:

DEVSEQ:VENDOR=Maxtor,SEQ=OverWriteDrive_SATA.seq

This tells AME that out of all the devices listed in STEP 2, any one of them which is made by "Maxtor", that AME is to run test sequence file "OverWriteDrive_SATA.seq" against them.

One of the advanced features of AME is that it can run two completely different test sequences simultaneously, one test sequence to one set of drives, and another test sequence to another set of drives, but done SIMULTANEOUSLY!!

Other DEVSEQ usages are as follows:

DEVSEQ:PRODUCT=MX00486,SEQ=OverWriteDrive.seq (change MX00486 to whatever your product is)

DEVSEQ:HBA=17,SEQ=OverWriteDrive.seq (change 17 to whatever your HBA number is)

IMPORTANT: Use of the "DEFAULT" keyword. You can tell AME a default test sequence to run against a drive. Here's an example:

DEVSEQ:VENDOR=DEFAULT,SEQ=OverWriteDrive.seq

What the above tells AME is that no matter what the VENDOR is of a drive, to use sequence "OverWriteDrive.seq"

Here's an example of an AMEConfig.txt file:

PATHTOLOGFILE:d:\workdirv900\AMELogfiles

DEVSEQ:VENDOR=Seagate,SEQ=OverWriteDrive_SCSI.seq

DEVSEQ:VENDOR=Maxtor,SEQ=OverWriteDrive_SATA.seq

STEP 4: Launch AME

The parameters to launch AME from a command prompt window or a batch file are as follows:

AME.exe Configuration_File_Name List_of_Devices_To_Test

Notice there are just two command-line parameters to pass AME.exe

Here's an example from STEP 2:

Example 1: AME.exe AMEConfig.txt device=3-5-0,6-0-0